Aircraft Reply and Interference Environment Simulator (ARIES) Hardware Principles of Operation: Volume I

Edward Mancus

October 1989

DOT/FAA/CT-TN88/4-I

U.S. Department of Transportation
Federal Aviation Administration

Technical Center
Atlantic City International Airport, N.J. 08405

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/CT-TN88/4-I		

4. Title and Subtitle

AIRCRAFT REPLY AND INTERFERENCE ENVIRONMENT SIMULATOR (ARIES) HARDWARE PRINCIPLES OF OPERATION: VOLUME I

5. Report Date
October 1989

6. Performing Organization Code

7. Author(s)
Edward Mancus

8. Performing Organization Report No.
DOT/FAA/CT-TN88/4-I

9. Performing Organization Name and Address
Federal Aviation Administration
Technical Center
Atlantic City International Airport, New Jersey 08405

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
U.S. Department of Transportation
Federal Aviation Administration
Technical Center
Atlantic City International Airport, New Jersey 08405

13. Type of Report and Period Covered
Technical Note

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The Aircraft Reply and Interference Environment Simulator (ARIES) makes possible the performance assessment of the Mode Select (Mode S) sensor under its specific maximum aircraft load. To do this, ARIES operates upon disk files for traffic model and interference to generate simulated aircraft replies and fruit, feeding them to the sensor at radio frequency.

Support documentation for ARIES consists of:

ARIES Hardware Maintenance Manual: Volume I (DOT/FAA/CT-TN88/3)
Appendixes of the Hardware Maintenance Manual: Volume II
ARIES Hardware Principles of Operation: Volume I (DOT/FAA/CT-TN88/4-I)
Appendixes of the Hardware Principles of Operation: Volume II
ARIES Software Principles of Operation (DOT/FAA/CT-TN87/16)
ARIES Software User's Manual (DOT/FAA/CT-TN88/15)

This document, the ARIES Hardware Principles of Operation, Volume I, explains the theory of operation of the ARIES special purpose hardware designed and fabricated at the Federal Aviation Administration Technical Center. Each hardware device is discussed. Functional block diagrams, signal timing diagrams, and state timing diagrams are included where appropriate.

17. Key Words		18. Distribution Statement
ARIES	Mode S Beacon System	Document is on file at the Technical Center Library, Atlantic City International Airport, New Jersey 08405
Fruit Replies	Modeled Replies	
Air Surveillance Radar (ASR-9)		
Common Digitizer (CD-2)		

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	276	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF TABLES

EXECUTIVE SUMMARY

The Aircraft Reply and Interference Environment Simulator (ARIES) is a stand-alone target simulation system designed and fabricated at the Federal Aviation Administration (FAA) Technical Center to analyze the performance characteristics of the Mode Select (Mode S) Beacon System sensor operating within capacity and high density beacon environments. A total of four ARIES systems have been fabricated to support the development of the Mode S sensor during its production phases. In addition, the ARIES will be used to support acceptance testing to be conducted by the FAA.

Volume I of the Hardware Principles of Operation Manual introduces the user to the special purpose hardware developed for the ARIES. This manual gives the user an overview of its functions and capabilities. ARIES hardware operations are described in sufficient detail, with the assistance of block diagrams, logic diagrams, and flow charts, for a user familiar with analog and digital hardware to maintain the system with the use of the ARIES diagnostic package.

1. INTRODUCTION.

This manual is a general user guide describing the principles of operation of the Aircraft Reply and Interference Environment Simulator (ARIES) special purpose hardware. This hardware was designed and fabricated by Federal Aviation Administration (FAA) Technical Center personnel. The hardware description of off-the-shelf equipment employed in ARIES, such as computer systems, printers, disk drives, etc., is covered by the appropriate manufacturer's hardware manuals and will be addressed only at the level necessary to provide an overall understanding of the ARIES as a system.

A general background for the purpose of ARIES is given in section 2. Reference documents and other related manuals are listed in section 3. Capabilities of the ARIES system are summarized in section 4. The physical and functional layout of the ARIES system is described in section 5, and the principles of operation of each of the special purpose hardware devices used in ARIES are explained in section 6. These sections, along with the appendixes of this manual, provide enough background information for a person experienced in digital and analog hardware design to use hardware drawings to perform repair and/or calibration on the ARIES special purpose hardware when necessary.

2. BACKGROUND.

The FAA is currently improving the National Airspace System (NAS) to keep pace with the projected traffic growth through the year 2000. One major element in this improvement effort is the deployment of a new secondary radar system referred to as the Mode Select (Mode S) Beacon System. The Mode S sensor works in conjunction with airborne transponders to obtain information on aircraft identity and altitude, and serves as the basis for digital communications between the ground system and the aircraft.

In 1986, the FAA awarded a contract for the procurement of 137 Mode S sensors to be installed in the more busier air traffic locations. These sensors will provide increased target capacity handling, better azimuth measurement precision, and reduced interference between sensors due to reduced interrogation rates as compared to the current Air Traffic Control Radar Beacon System (ATCRBS).

Due to the large target capacity of the new technological ground-based surveillance system, the FAA recognized that it was not possible to find a current beacon environment that is dense enough to fully test the Mode S sensor under heavy load conditions. Furthermore, a Mode S transponder environment does not exist currently, and so it is not possible to provide any significant loading of the Mode S specific functions. For this reason, and also for the desire to be able to repeat the identical test several times, an environment simulator was necessary to appear to the Mode S sensor under test as a dense beacon environment plus a dense interfering environment typical of what might be encountered in the future. These were the primary reasons for the development and fabrication of the ARIES system.

The ARIES is designed to simulate a radar beacon environment of up to 700 aircraft equipped with transponders plus high rates of interfering beacon replies "fruit." So using this system, the Mode S sensor can be fully tested under heavy target loads.

Four ARIES systems will be provided as government furnished equipment to the Mode S sensor production contractors, Westinghouse, and Unisys. The systems will be used to simulate, at the radio frequency (RF) level, capacity and high density aircraft and interference loading environments to the Mode S sensor during tests required by the following paragraphs of FAA-E-2716:

a. 4.3.2 Contractor preliminary tests
b. 4.3.3 Design Qualification tests
c. 4.3.4 Type tests
d. 4.4.2.3 Hardware/Software Integration tests

This makes the ARIES an important link in the successful implementation of the Mode S System.

3. RELATED DOCUMENTS.

a. Goon, M. and Spencer, D. A., Aircraft Reply and Interference Environment Simulator (ARIES), FAA-RD-78-96, March 22, 1979.

b. Mode Select Beacon System (Mode S) Sensor Specifications, FAA-E-2716, March 24, 1983.

c. U.S. National Standard for the IFF Mark X (SIF) Air Traffic Control Radar Beacon System (ATCRBS), FAA Order 1010.51A, March 8, 1971.

d. U.S. National Standard for the Mode Select Beacon System, FAA Order 6365.1A, January 3, 1983.

e. The Aircraft Reply and Interference Environment Simulator (ARIES) Hardware Maintenance Manual, DOT/FAA/CT-TN88/3 Volume I and II, May 1988.

f. The Aircraft Reply and Interference Environment Simulator (ARIES) Software Principles of Operation, DOT/FAA/CT-TN88/15, April 1988.

g. Mode S to Common Digitizer (Model 2) Interface Control Document, TM-PA-0018/074/02A, March 23, 1987.

h. ASR-9 External Interface Control Document for the ASR-9 C&I Processor to Mode S Sensor Interface and the Mode S to ASR-9 SP Interface, DTFA01-83-C-20027, September 24, 1984.

4. ARIES SYSTEM CAPABILITIES.

The ARIES is capable of simulating a high density radar beacon environment of up to 700 aircraft. Any mix of Mode S and ATCRBS targets is possible. The target environment can be patterned to simulate any capacity and high density aircraft environment necessary to establish Mode S sensor performance with respect to specified requirements. ARIES can simulate these target environments in either single or dual (back-to-back) antenna modes. In addition to simulating the maximum number of targets handled by the Mode S sensor, ARIES is capable of handling the target bunching and data link peaking requirements specified for the Mode S sensor, which is capable of handling:

a. 250 targets within a 90^{o} quadrant.

b. 50 targets within an 11.25^{o} sector, for not more than four consecutive sectors.

c. 32 targets within a 2.4^{o} wedge.

Along with the modeled target environment, ARIES is capable of generating a simulated fruit environment consisting of both ATCRBS and Mode S interfering beacon replies. The arrival times of fruit replies are not based on the traffic model. To do this would require modeling the nearby interrogators that cause these interfering replies to be generated. Instead, fruit is modeled as a random process with Poisson statistics.

For both the modeled transponder and fruit replies, ARIES provides the necessary signals to accurately simulate the monopulse offboresight angle. Also, an omnidirectional signal is provided so that sidelobe replies can be simulated. These signals are connected to the Mode S sensor via an RF interface to ARIES. Inside the sensor they are summed with similar signals from the sensor's own antenna. This allows a simulated environment to be superimposed on a live environment if desired.

In addition to the beacon data, ARIES provides simulated digitized radar data in the output format of an Airport Surveillance Radar model 9 (ASR-9) or a Common Digitizer model 2 (CD-2). The radar targets correspond to the simulated beacon targets. The reported coordinates are those that would be seen by a primary radar whose antenna rotates with the beacon antenna about the same axis.

Each of the above mentioned features of ARIES is described in more detail in the following sections. Figure 4-1, an overall view of the ARIES hardware, will be useful in understanding those paragraphs describing the implementation of these features.

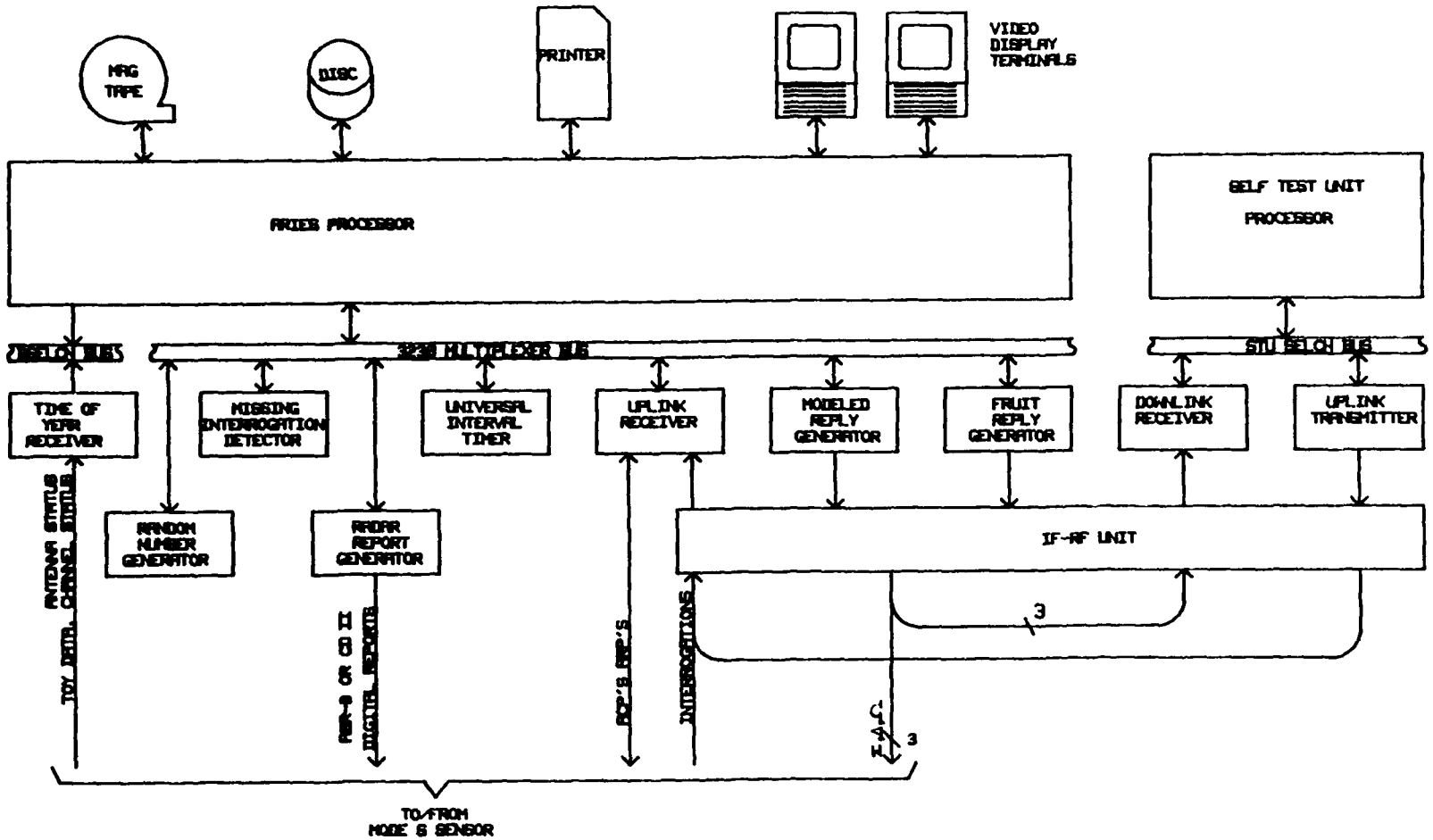


FIGURE 4-1. ARIES HARDWARE BLOCK DIAGRAM

4.1 MODELED REPLY GENERATION.

ARIES is capable of simulating both ATCRBS transponder-equipped aircraft and Mode S transponder-equipped aircraft whose characteristics are defined within a traffic model file stored on the system disk. Each logical record specifies the position and velocity of one aircraft at a particular time, as well as such items as its altitude, its identity code, and those items of state information to be mentioned in sections 4.1.1 and 4.1.2. These records are stored in increasing time order. The ARIES computer reads these records from the disk and when the current system time matches the time tag of the model record, that record is used to update the track file entry for that target or to start a new entry if this is the first appearance of the target. Typically, targets are updated periodically by the traffic model file at 4-second intervals. More frequent updates from the traffic model can create more precise flight paths. For more detail on the contents of the traffic model file and how to generate new models, refer to the ARIES Software Principles of Operations Manual.

ARIES receives interrogations from the Mode S sensor at 1030 megahertz (MHz) and processes them by the receiver circuitry. The receiver transfers a data block to the computer giving the type of interrogation, the time-of-arrival, the boresight azimuth of the antenna, whether the interrogation was received by the front antenna or the back antenna, and any data content. The azimuth data is provided by the azimuth decoder/simulator which can operate in either of two modes. In the decoder mode, azimuth change pulses (ACPs) and azimuth reference pulses (ARPs) are received from the sensor's antenna system, and used to increment and reset an azimuth register, respectively. The ACPs and ARPs are also retransmitted, as received, allowing the option of inserting the ARIES into these lines between the antenna and the sensor. In the simulator mode, ACPs and ARPs are produced by ARIES and sent to the sensor. ACPs and ARPs from the antenna, if the antenna is connected, are ignored. Thus, the ARIES can be run with a sensor that is not connected to an antenna system. The decoder mode must be used in the case where live data from the sensor's own antenna and the simulated data from ARIES are to be superimposed.

In the computer, the response depends on the type of interrogation. If it is an ATCRBS/All-Call interrogation, all ATCRBS transponders and unlocked Mode S transponders that are in the antenna beam are located in the track file resident in computer memory. If it is a Roll-Call interrogation, the appropriate track record is located by means of its Mode S address. Reply data blocks are generated for each such target. These specify the time of reply, the reply type, the reply power level, the simulated offboresight angle, and the data content. This data is transferred to the modeled reply generator circuitry. This generates the appropriate reply waveforms at 1090 MHz for acceptance by the Mode S sensor. The parity check encoding for Mode S replies are performed by the digital reply generators. Since there are three independent reply generators, up to three overlapping replies can be simulated. The signals of all these are summed together in the intermediate frequency (IF) combiner, then stepped up to the RF level and output to the sensor.

4.1.1 Simulated ATCRBS Transponder Behavior.

Simulated ATCRBS transponders are capable of responding to the following interrogation types:

a. Mode A ATCRBS/Mode S All-Call
b. Mode A ATCRBS-Only All-Call
c. Mode C ATCRBS/Mode S All-Call
d. Mode C ATCRBS-Only All-Call
e. Mode A
f. Mode C
g. Mode 2

The simulated ATCRBS transponders are capable of generating mode A, mode C and, if designated as equipped, mode 2 replies with any data pattern including the special position indicator (SPI) pulse, as defined in the ATCRBS National Standard. The simulated transponder will reply to the above mentioned interrogation types with the appropriate reply type, i.e., a mode A reply for a mode A interrogation unless one or more of the following conditions exist:

a. When the transponder azimuth lies outside of the antenna beam. For this purpose, the antenna beam edges are defined by the points where absolute value of the DIFF divided by the SUM is equal to 2.

b. When a fourth reply generator is required.

c. When the specified reply probability of a transponder is less than one, and the transponder randomly fails to reply in accordance with that reply probability.

d. When the slant range of the transponder is less than 1 nautical mile (nmi).

e. When the interrogation mode is mode 2 and the transponder is not designated as being mode 2 equipped.

4.1.2 Simulated Mode S Transponder Behavior.

Simulated Mode S transponders are capable of responding to the following interrogation types:

Mode A ATCRBS/Mode S All-Call

Mode C ATCRBS/Mode S All-Call

Mode S short interrogation (56 bits)

Mode S long interrogation (112 bits)

The Uplink Format (UF) field for the Mode S interrogation must have a code of 4 (surveillance, altitude), 5 (surveillance, identity), 20 (Comm-A, altitude), 21 (Comm-A, identity), or 24 (Comm-C). Mode S-only All-Calls (UF code 11) is ignored by ARIES.

The simulated Mode S transponders are capable of generating Mode S short (56 bits) and long (112 bits) replies with any data pattern as defined in the Mode S National Standard. The simulated transponder will respond with the appropriate reply (Downlink Format (DF) field of 4 (surveillance, altitude), 5 (surveillance,

identity), 11 (Mode S-only All-Call), 20 (Comm-B, altitude), 21 (Comm-B, identity), or 24 (Comm-D) unless one or more of the same conditions mentioned for ATCRBS to reply holds. The contents of the control and protocol fields (first 32 bits) of each reply is determined by ARIES based on both information contained in the interrogation and the traffic model. A summary of these function are listed below:

a. Lockout of simulated transponders is supported for nonselective All-Call lockout protocol and multisite All-Call lockout protocol.

b. The flight status (FS) field of a Mode S reply is defined by the traffic model. The field may take on any value.

c. The capability (CA) field of a Mode S All-Call reply is defined by the traffic model. Also, the extended capability report is defined by the traffic model.

d. Reporting of transponder altitude or identity code.

e. Both air-initiated and ground-initiated Comm-B standard and multisite protocol is simulated. The contents of the Comm-B message (MB) field is defined by the traffic model. The MB field of all other Comm-B replies contain a constant determined by the traffic model.

f. Both the uplink and downlink extended length message standard and multisite protocol is supported. The Comm-C message (MC) field is ignored except as required to support the protocols. The Comm-D message (MD) fields contain a constant determined by the traffic model except as required to support the protocols. Downlink requests are initiated under control of the traffic model and any number of downlink segments up to 16 can be specified.

A final characteristic of the simulated transponders (ATCRBS and Mode S) that can be controlled via the traffic model is the reply probability. Each simulated transponder has its own reply probability value, and this can be used to simulate various round-reliability conditions. The hardware random number generator shown in figure 6.7-1 is used to determine whether or not a target will reply.

4.2 FRUIT REPLY GENERATION.

ARIES is capable of generating both ATCRBS and Mode S fruit replies. These replies, combined with modeled target replies, are coupled at the RF level and output to the Mode S sensor.

4.2.1 ATCRBS Fruit Replies.

ATCRBS fruit replies are generated at an average rate between 1,000 to 50,000 replies per second. The fruit rate can be independently specified within each sector of a scan, a sector being a 11.25^{o} wedge. In addition, the fruit rate can be updated every 10 seconds.

The capability of ARIES to generate high fruit rates is required to test the performance of the Mode S sensor reply processing circuitry at the interference level at which it is specified to operate. Because of the high rates required, the entire reply generation process is performed in hardware.

The computer updates the fruit environment based on a defined fruit definition file. This file contains the ATCRBS fruit parameters, as well as the Mode S fruit parameters to be discussed in the next section, stored in a time ordered sequence. The specific ATCRBS fruit parameters are (1) the average fruit rate desired, (2) the fraction of the replies that are to appear in the mainbeam of the antenna as opposed to the sidelobes, (3) the fraction of replies that are to have a fixed code, and (4) the specified fixed ATCRBS reply code. The later two items allow a particular code to be emphasized in the otherwise randomly generated stream of reply codes. The likely candidate for emphasis might be, for example, the nondiscrete code 1200.

The fruit inter-arrival times match those expected from a Poisson process, i.e., the probability that any given inter-arrival interval lies between t and t-dt is given by;

$$P(t) = Ae^{-At}\,dt$$

where A is the average arrival rate as specified by the computer. The allowable range for A is from 1,000 to 50,000 fruit per second. However, there are only three reply generators, so no more than three overlapping replies can be simulated even though this is a high probability event at a fruit rate of 50,000 fruit per second. The number of ATCRBS fruit reply generators provided was chosen to match the Mode S sensor's ability to decode a legitimate reply in the presence of up to three overlapping fruit replies.

Reply parameters, such as offboresight angle, power, and code bits are determined by pseudo-random generators. The term pseudo-random is used to indicate that the sequence of replies can be repeated exactly by initializing the fruit generator (i.e., the sequence is deterministic) although the reply statistics are designed to match those of certain random distributions.

The monopulse angle of the fruit replies is uniformly distributed across the entire range of offboresight angles that are generated by the ARIES.

The fraction of mainbeam fruit, as opposed to sidelobe fruit, is specified by the computer according to a parameter stored in the fruit definition file as mentioned earlier. The range of mainbeam/sidelobe fruit can be specified from 0 to 1 in increments of 1/255. For mainbeam fruit, the SUM channel signal level as measured at the sensor's RF port is randomly chosen according to the distribution;

$$P_{ml} = (-20 - 20*\log r_{ml})\ \text{dBm}$$

where r_{ml} is a random integer uniformly distributed between 1 and 100 nmi. For sidelobe fruit, the equivalent distribution is;

$$P_{sl} = (-55 - 20*\log r_{sl})\ \text{dBm}$$

where r_{sl} is a random integer uniformly distributed between 1 and 32 nmi. The sidelobe replies are simulated by attenuating the SUM and DIFF signals below the OMNI signal. The sidelobe attenuation is set between 20 decibel (dB) and 36 dB, adjustable in steps of 1 dB. The resolution for both power levels is 1 dB.

The reply codes, other than the specified fraction that is generated with a fixed code, are generated according to a distribution that roughly adjusts the probability of each of the ATCRBS data bits according to the likelihood of their being set in real fruit replies. Thus, not all reply codes are equally likely. For example, certain bit patterns are illegal in mode C replies, and these are therefore less likely to occur, although they are legal mode A patterns. Similarly, patterns corresponding to the mode C replies of very high flying aircraft are less likely than others. The X bit and the SPI bit are fixed at 0.

4.2.2 Mode S Fruit Replies.

Mode S fruit replies are generated at an average rate between 10 to 640 replies per second. The fruit rate can be independently specified within each sector and updated every 10 seconds, same as the ATCRBS fruit rate.

The computer updates the fruit environment based on parameters stored in a defined fruit definition file. The specific Mode S fruit parameters are (1) the average fruit rate desired, (2) the fraction of the replies that are to appear in the mainbeam of the antenna as opposed to the sidelobes, and (3) the fraction of replies that are surveillance replies as opposed to Comm B or Comm D replies. In the later item, the allowable range is 0 to 1 in steps of 1/16.

The reply generation process is performed entirely in hardware similar to that used to generate ATCRBS fruit replies. The inter-arrival times for Mode S fruit is exponentially distributed to match those expected from a Poisson process similar to the ATCRBS inter-arrival time given by;

$$P(t) = Be^{-Bt}\,dt$$

where B is the average arrival rate as specified by the computer. The allowable range for B is 10 to 640 fruit per second.

In the generation of the offboresight angle, the fraction of replies in the mainbeam as opposed to the sidelobes and the SUM power distributions are performed in the same way as defined for the ATCRBS fruit.

The reply data contents are generated according to a distribution of likely values. The values generated for the following fields: Downlink Format (DF:5), Flight Status (FS:3), Downlink Request (DR:5), Capability (CA:3), Altitude Code (AC:13), and Identity (ID:13) are restricted to defined or legal values and given equal probability. For example, the contents of the Downlink Format field will contain a legal value, that is, if a short reply is generated, its DF field will specify that the reply is a defined short reply. The contents of the remaining data fields are generated by a pseudo-random number generator.

4.3 RADAR REPORT GENERATION.

The ARIES is capable of generating simulated primary radar data from two different primary radars; the ASR-9 and the CD-2. Only one type of primary radar data is simulated during any simulation run. A primary radar report is generated on each scan for each active target specified in the traffic model and is subject to a radar blip/scan ratio specified by the ARIES operator. The blip/scan ratio can have a value of 0 to 1 with a resolution of 1/31. The intermediate values result in the specified system average blip/scan ratio which is used in conjunction with a hardware random number generator to determine on a pseudo-random basis whether a given target will have a radar report on a given scan.

Radar reports are transmitted to the Mode S sensor just after the target has left the antenna beam but within the specified time limit of 3/64 of a scan from the time of the reported azimuth for sensor radar processing. The range and azimuth coordinates used are exactly those obtained from the traffic model as extrapolated to the time of boresight crossing, but converted to the appropriate units. This data is transferred to the radar report generator which outputs the data to the sensor using the bit-oriented Advanced Data Communication Control Procedure (ADCCP) specified in the following documents: (1) "Mode S to Common Digitizer (model 2) Interface Control Document," and (2) "ASR-9 External Interface Control Document for the ASR-9 C&I Processor to Mode S Sensor Interface and the Mode S to ASR-9 SP Interface."

In addition, a Search Real Time Quality Control (RTQC) report is transmitted at the 180^{o} azimuth mark and one status report is transmitted at the 0^{o} azimuth mark each scan.

The interface is capable of transferring the numbers of reports normally anticipated from the ASR-9 or CD-2 at the rate of 500,000 bits per second.

4.4 SELF-TEST CAPABILITIES.

ARIES is a relatively complex system and is being used to test an even more complex system. It is essential, therefore, that there be some reliable way of confirming that the ARIES hardware is functioning correctly prior to and following simulation runs. This verification must be independent of the Mode S sensor.

The certification of standard off-the-shelf equipment, such as processors, tape drives, disk units, etc., is accomplished using the diagnostics provided by the manufacturer or supplier. These diagnostics are available to be run by the ARIES operator. The certification of the special purpose hardware designed for ARIES is accomplished using the ARIES hardware maintenance software package developed by the FAA Technical Center. This package is described in detail in the "ARIES Hardware Maintenance Manual."

Diagnostic circuitry is incorporated with each of the special digital devices to support several modes. These modes by and large fall into the category of writing data to the device and then reading it back. In some cases, the device command bits affect which data is read back and so their functioning can be tested as well. Also, each digital device containing a bit-sliced microprocessor has stored in their microcode (firmware) a set of diagnostic support routines that allow the testing of hardware not directly accessible to the computer. These routines, along with the description of the microcode set used, are described in appendix B (Volume II - Hardware Principles of Operation).

It is not possible to test all the digital logic and none of the RF or IF circuitry in the above mentioned fashion. This is the purpose of the Self Test Unit (STU) included in the ARIES. The STU provides for the complete testing of the interrogation detection/processing and the reply generation/transmission circuitry of the ARIES. It is also useful for system checkout purposes. The STU can be programmed to generate any interrogation pattern and with the azimuth simulation capability of the ARIES, allow the ARIES (both hardware and software) to run in its normal simulation mode without connecting to a Mode S sensor.

The STU consists of its own dedicated computer and two special digital devices, the Uplink Transmitter and the Downlink Receiver. The Uplink Transmitter is capable of generating at 1030 MHz all of the interrogation modes and data patterns that are processed by the ARIES interrogation receiver. The interrogations are coupled into the normal receiver path with the interrogations received from the Mode S sensor. The interrogation type and its time of transmission are under computer control. The Downlink Receiver is capable of demodulating and detecting all of the reply types and sampling all of the reply data bits that are transmitted by the ARIES reply generators. When a reply bracket or a preamble is detected by the STU, an amplitude sample of the SUM and DIFF or OMNI channels is taken and the data bits sampled. The choice between sampling the DIFF or OMNI channels is determined by the computer. Decoding of the Mode S parity check is performed in hardware and the time of the reply is sampled with a same resolution specified for the reply generators. All of the information gathered on the reply is then fetched by the STU's computer for comparison with anticipated results. Both the modeled reply generators and he fruit reply generators can be tested by this method. The fruit reply generators require the use of a special diagnostic mode whereby the ARIES's computer specifies the reply parameters and data, rather than having these randomly generated.

5. ARIES EQUIPMENT.

Before describing in detail the operations of the ARIES special purpose hardware, a brief description of the overall ARIES equipment will be presented. The discussion will cover the equipment making up the ARIES and its physical layout.

The ARIES equipment consists of interrogation receiving circuitry, reply generation circuitry, self test circuitry, two computers, and associated peripherals to control the system. This equipment is housed in three standard cabinets as shown in figure 5-1. A hardware overview of these equipment, along with their salient characteristics, will be presented in the rest of this section.

The peripherals used by the ARIES system consists of a disk drive, a dual density magnetic tape unit, two video display terminals, and a printer. The salient characteristics of these peripherals are listed in table 5-1. The disk drive and the magnetic tape unit are housed in one cabinet, referred to as the peripheral cabinet as shown in figure 5-2. The remaining peripherals are stand-alone units.

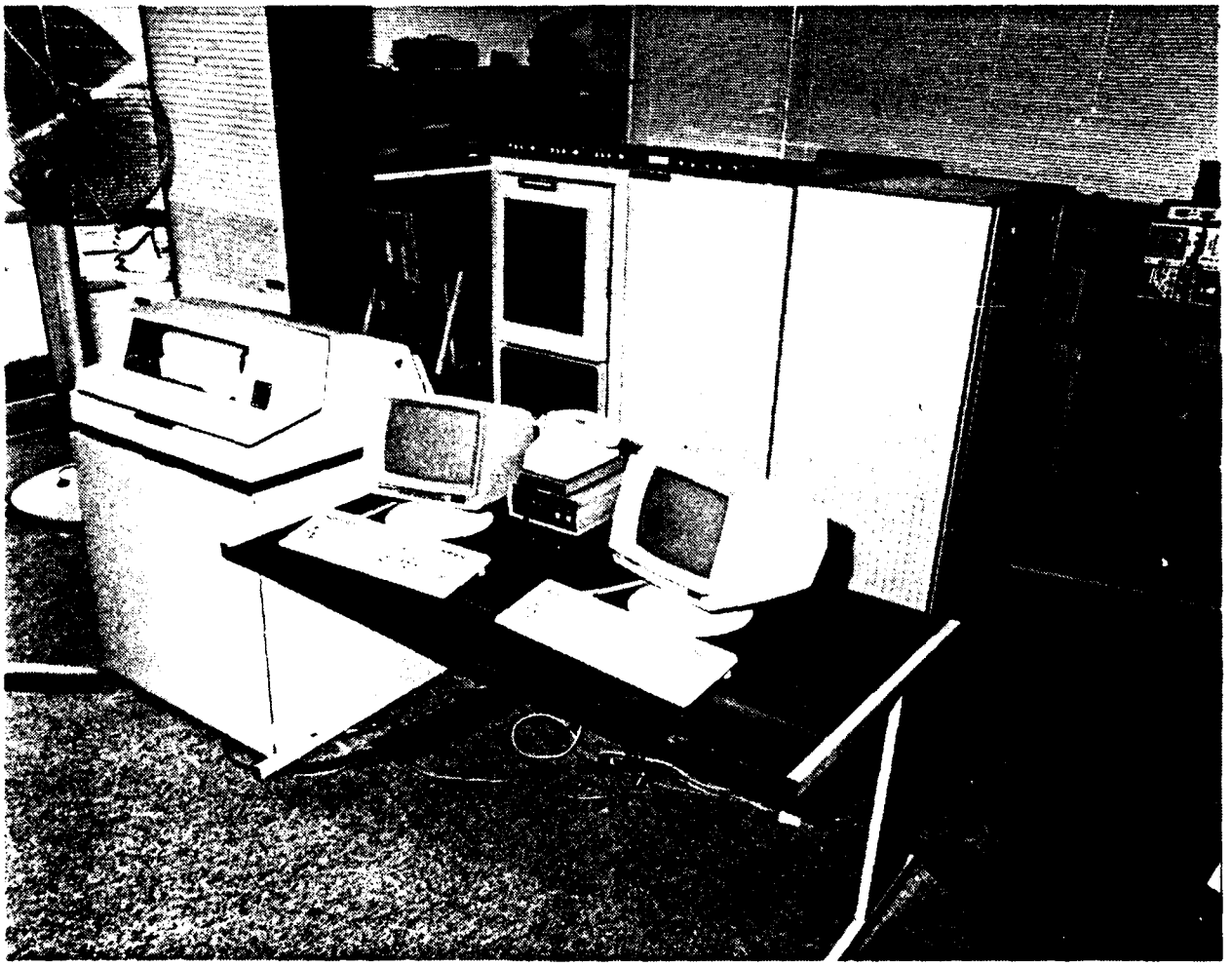


FIGURE 5-1. ARIES EQUIPMENT OVERVIEW

TABLE 5-1. SALIENT CHARACTERISTICS OF PERIPHERALS USED IN ARIES

PERIPHERAL	SALIENT CHARACTERISTICS
Magnetic Tape Unit	Quantity one per system 75 ips, 9 track Dual (800/1600 cpi) recording density 1/2 inch, 2400 foot (10.5 inch) reel 120,000 cps transfer rate - 1600 cpi
Disk Storage Unit	Quantity one per system 300 Megabyte storage 1.2 Megabyte data transfer rate
Video Display Unit	Quantity two per system 80 characters per line 24 lines per screen 128 ASCII character set RS232C standard current mode interface
Line Printer	Quantity one per system 300 lines per minute, 132 columns/line USASCII 96 character set

The input/output (I/O) bus chassis and its associated power supply is housed in the peripheral cabinet as well. This chassis contains the intelligent disk controller board, the magnetic tape controller board, a buffer selector board for direct memory address operations, and two I/O bus switch B-boards (The I/O bus switch A-boards that works in conjunction with these boards are discussed later.) The I/O bus switch consolette which selects which computer has access to the magnetic tape unit and the disk drive is located on top of the cabinet.

Two computer systems are housed in a second cabinet, referred to as the processor cabinet (see figure 5-2). The first computer system (Concurrent Computer Corporation 3230 32-bit computer), with 3 megabytes of random access memory (RAM), 4 kilobytes high-speed writable control store (WCS) memory, and with high-speed interface capability, interfaces with and controls the ARIES components. The second computer system (Concurrent Computer Corporation 3205 32-bit minicomputer), with 1 megabyte of RAM, interfaces with and controls the STU components. Each system is housed in their own chassis but share a common +5-volt power supply. Each computer is independent of the other with only one RS-232C communication link setup between them for the purpose of supporting maintenance functions (refer to appendix E of the ARIES Hardware Maintenance Manual for further information).

Two nonstandard boards are housed in the STU computer chassis. Each is a special purpose device used by the STU to support verification testing of ARIES hardware and software functions. These boards are the Uplink Transmitter and the Downlink Receiver boards. They are the only FAA Technical Center fabricated boards not located in the third cabinet to be discussed later.

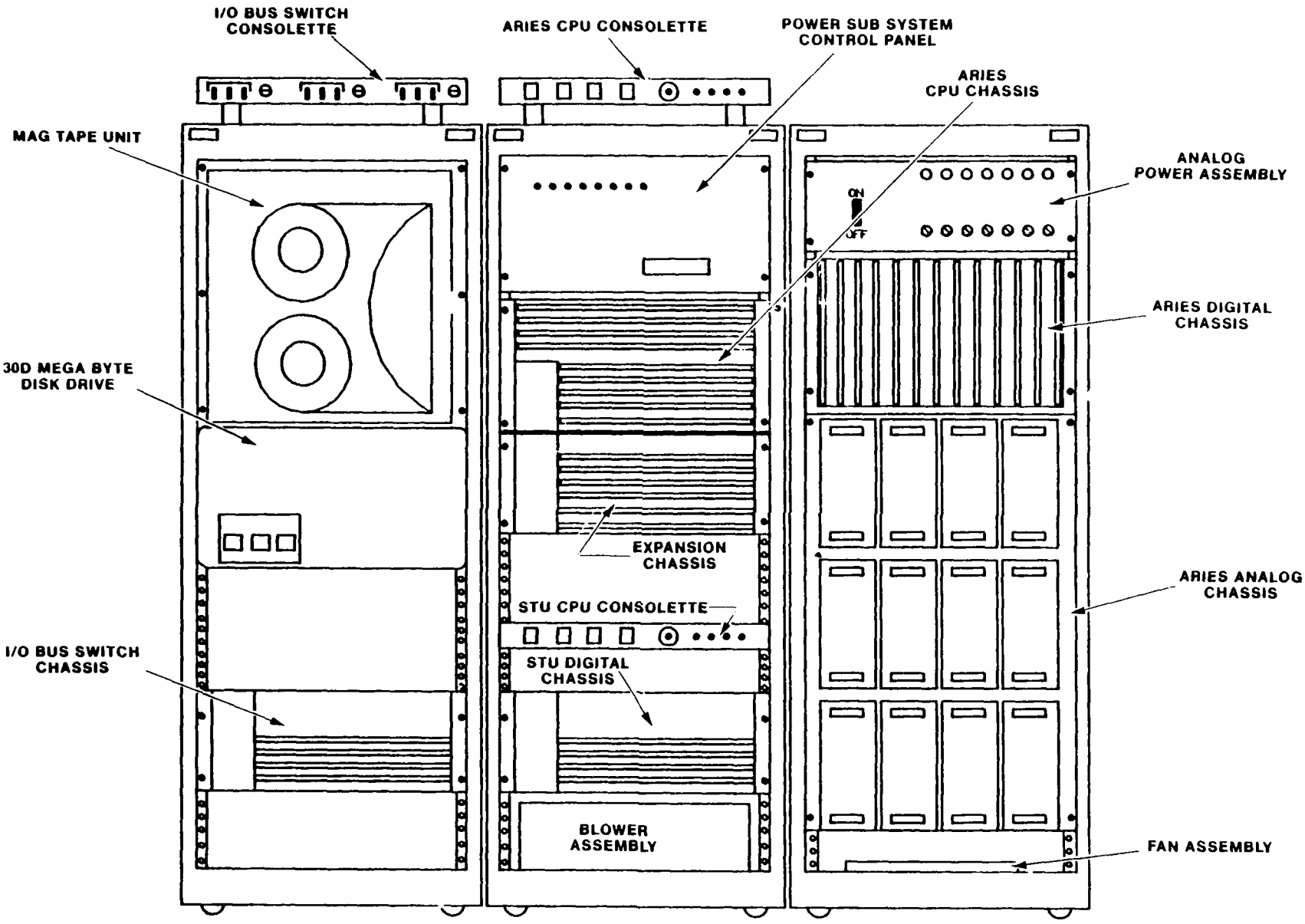


FIGURE 5-2. CABINET LAYOUT

Three +5-volt 150 Ampere switching power supply units are located at the top of the processor cabinet. One of these units is dedicated to the 3-megabyte memory of the ARIES computer. The remaining two power units, wired together to produce a 300 Ampere supply, provide power to the ARIES computer chassis, the STU computer chassis, and the ARIES digital chassis located in the third cabinet. Not shown in figure 5-2 is a battery backup unit for the main memory of the ARIES computer if power should be lost.

Each computer chassis contains a I/O bus switch A-board. Each A-board works in conjunction with an associated B-board located in the I/O bus chassis housed in the peripheral cabinet. When an A-board and its associated B-board are selected via the I/O bus switch consolette, the desired computer has access to the disk drive and magnetic tape unit.

The third cabinet houses all of the special hardware developed by the FAA Technical Center to support required functions of the ARIES system with the exception of previously mentioned digital boards located in the STU computer chassis. This cabinet is referred to as the hardware expansion cabinet (see figure 5-2). This special purpose hardware is covered in detail in section 6.

Two chassis are housed in the hardware expansion cabinet; one digital and one analog. The digital chassis contains all of the ARIES special purpose devices mounted on 12 digital boards and 1 printed circuit board. The digital chassis is located directly beneath the analog power assembly. The board slot assignment of this chassis is given in table 5-2. Each digital board has the chip capacity of 322 16-pin standard integrated circuit (IC) chips. The analog chassis contains all of the analog circuitry for the ARIES and the STU mounted on 12 removable panels for easy maintenance when required. The analog chassis is located beneath the digital chassis. The layout of this chassis is shown in figure 5-3.

The analog power assembly contains all of the power voltage levels required by the analog circuitry and the video pulse quantizer (VPQ) board located in slot No. 1 of the digital chassis. The analog power assembly is located at the top of the cabinet to minimize the heat flow through the cabinet since the air flow is forced from the bottom to the top of the cabinet by a fan assembly located at the bottom.

A break point panel is located inside of the cabinet behind and between the digital chassis and the analog chassis. This connector panel serves as the demarcation point for all interface cables between the two chassis and the STU chassis. All of the digital connectors used on this panel are 37-pin D-type connectors. A layout of this panel is shown in figure 5-4 and the name of the connectors is identified in table 5-3.

TABLE 5-2. ARIES DIGITAL CHASSIS BOARD SLOT ASSIGNMENTS

SLOT	BOARD NAME	DEVICE
1	Video Pulse Quantizer (VPQ)	Video Pulse Quantizer (VPQ) Dual Channel Video Sampler
3	Time-of-Year Receiver (TOY)	Time-of-Year Receiver (TOY)
5	Reply Generator No. 3 (RGEN)	Modeled Reply Generator (MRG)
7	Reply Generator No. 1 and No. 2 (RGEN)	"
9	Modeled Reply Controller (MRC)	"
11	Uplink Receiver (RVCR)	Uplink Receiver Azimuth Decoder/Simulator (AZGEN)
13	Multiple Device (MULT)	ARIES System Clock Universal Interval Timer (UIT) Random Number Generator (RNG) Missing Interrogation Timer (MIT)
15	Radar Report Generator (RRG)	Radar Report Generator (RRG)
17	Fruit Reply Controller (FRC)	Fruit Reply Generator (FRG)
19	ATCRBS Random Process Generator (ARPG)	"
21	Mode S Random Process Generator (MRPG)	"
23	Reply Generator No. 1 and No. 2 (RGEN)	"
25	Reply Generator No. 3 and No. 4 (RGEN)	"

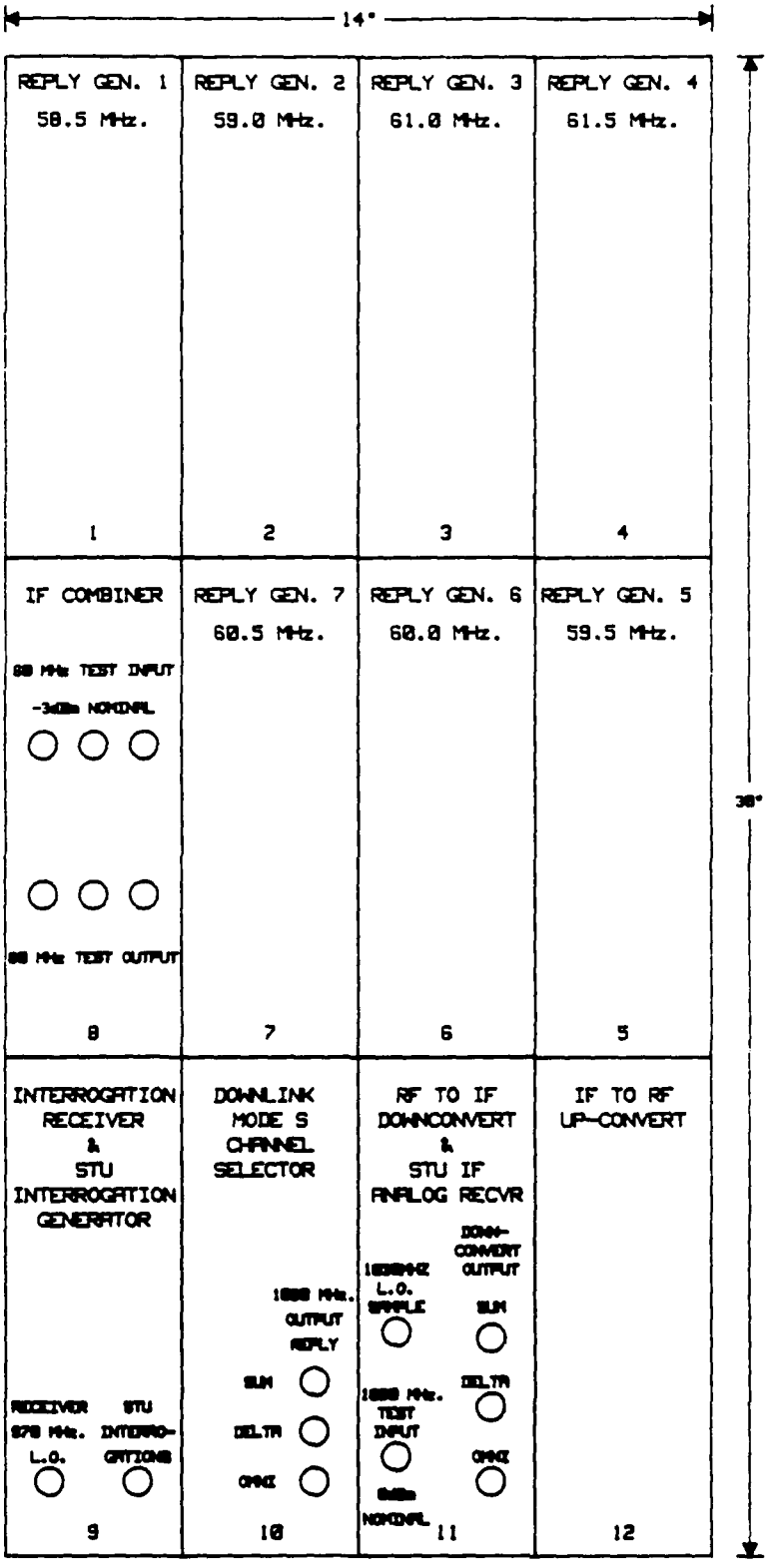


FIGURE 5-3. ARIES ANALOG CHASSIS LAYOUT

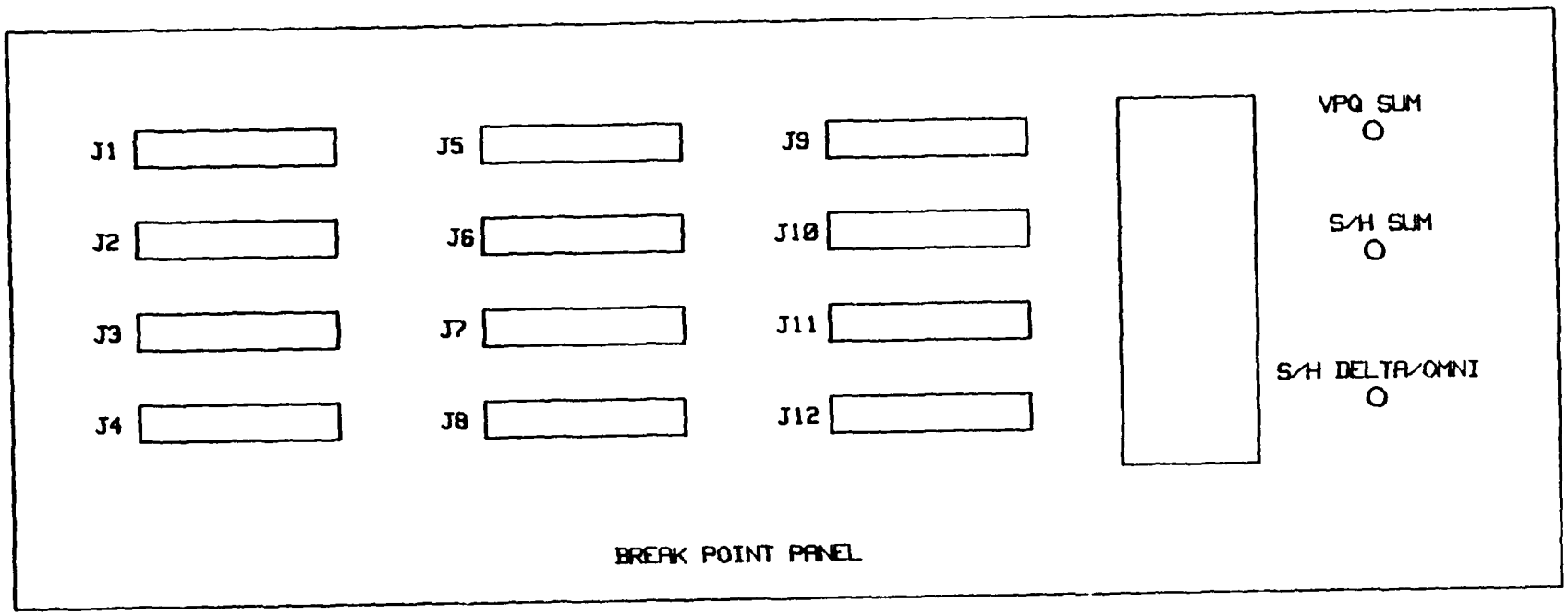


FIGURE 5-4. BREAK POINT PANEL LAYOUT

TABLE 5-3. BREAK-POINT PANEL CONNECTORS

CONNECTOR	TABLE
J1	Fruit Reply Generator No. 1
J2	Fruit Reply Generator No. 2
J3	Fruit Reply Generator No. 3
J4	Fruit Reply Generator No. 4
J5	Modeled Reply Generator No. 3
J6	Modeled Reply Generator No. 2
J7	Modeled Reply Generator No. 1
J8	Spare
J9	Video Pulse Quantizer
J10	Self Test Unit Connector No. 2
J11	Self Test Unit Connector No. 1
J12	Spare

The following three video cables also pass through this panel: one feeding SUM video to the VPQ, a second feeding SUM video to a video sampling channel, and a third feeding DIFF or OMNI video to the second video sampling channel from the RF to IF Convertor panel.

There is one other connector panel which serves as the junction point for all external interfaces to or from the ARIES system. This panel is referred to as the distribution panel and is located at the lower back of the hardware expansion cabinet. A layout of this panel is shown in figure 5-5 and the names of the connectors is identifies in table 5-4.

6. HARDWARE DESCRIPTION.

Figure 6-1 depicts the special purpose devices of the ARIES system and their interfaces to the Mode S sensor, the Mode S time-of-year (TOY) receiver, and the Mode S antenna. The principal special purpose devices interface to the computer are the Uplink Receiver, the Modeled Reply Generator (MRG), and the Fruit Reply Generator (FRG).

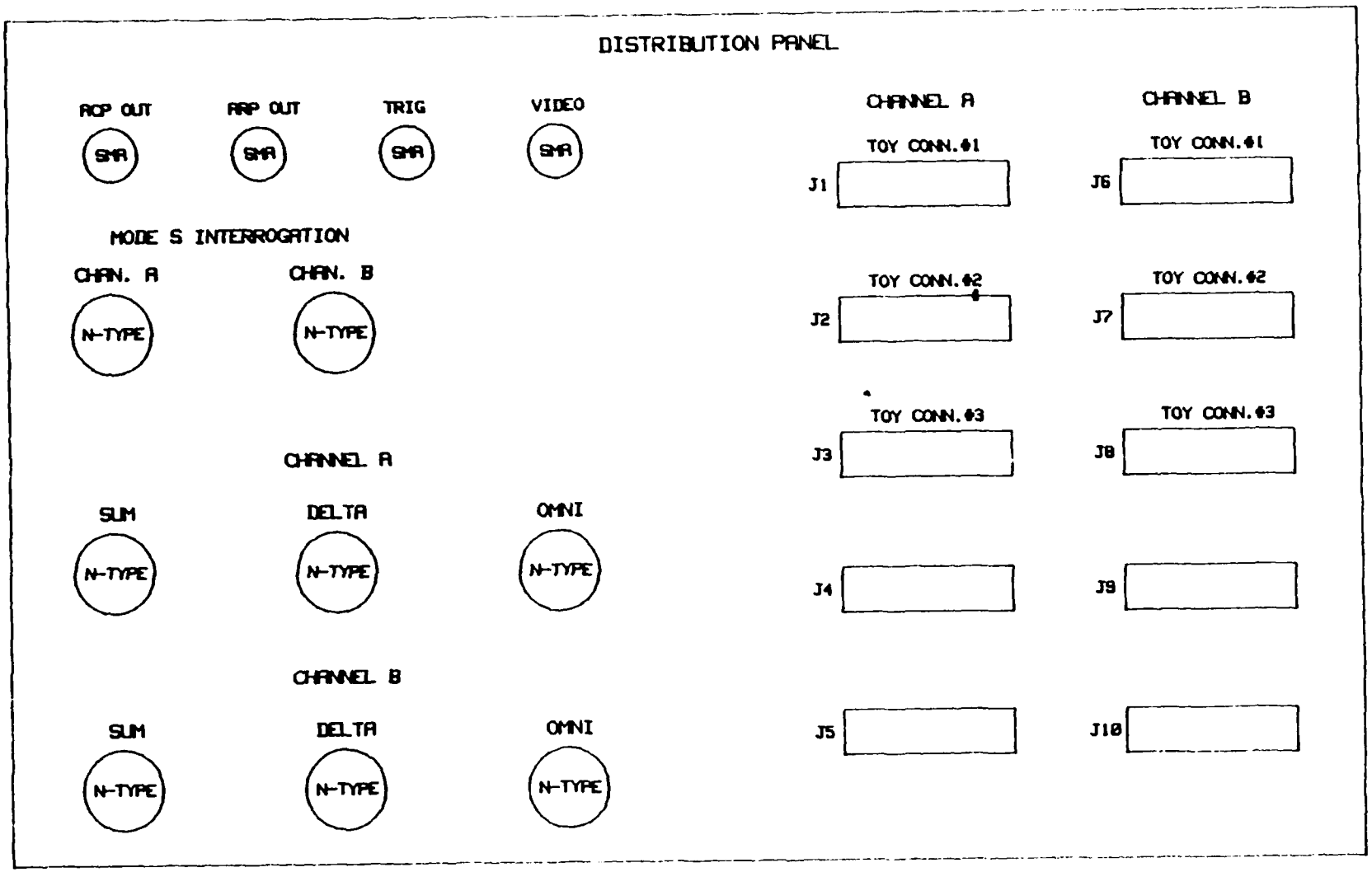


FIGURE 5-5. DISTRIBUTION PANEL LAYOUT

TABLE 5-4. DISTRIBUTION PANEL CONNECTORS

CONNECTOR	TYPE	LABEL
	BNC	ACP OUT (4096 ACPs)
	BNC	ARP OUT
	BNC	TRIG All-Call trigger generated on P3
	BNC	VIDEO (SUM)
	N	Mode S Interrogation Channel A
	N	Mode S Interrogation Channel B
	N	Channel A SUM
	N	Channel A DELTA
	N	Channel A OMNI
	N	Channel B SUM
	N	Channel B DELTA
	N	Channel B OMNI
J1	37-Pin D	Channel A TOY Connector No. 1
J2	37-Pin D	Channel A TOY Connector No. 2
J3	37-Pin D	Channel A TOY Connector No. 3
J4	37-Pin D	Channel A Radar Port Adapter
J5	37-Pin D	Spare
J6	37-Pin D	Channel B TOY Connector No. 1
J7	37-Pin D	Channel B TOY Connector No. 2
J8	37-Pin D	Channel B TOY Connector No. 3
J9	37-Pin D	Channel B Radar Port Adapter
J10	37-Pin D	Spare

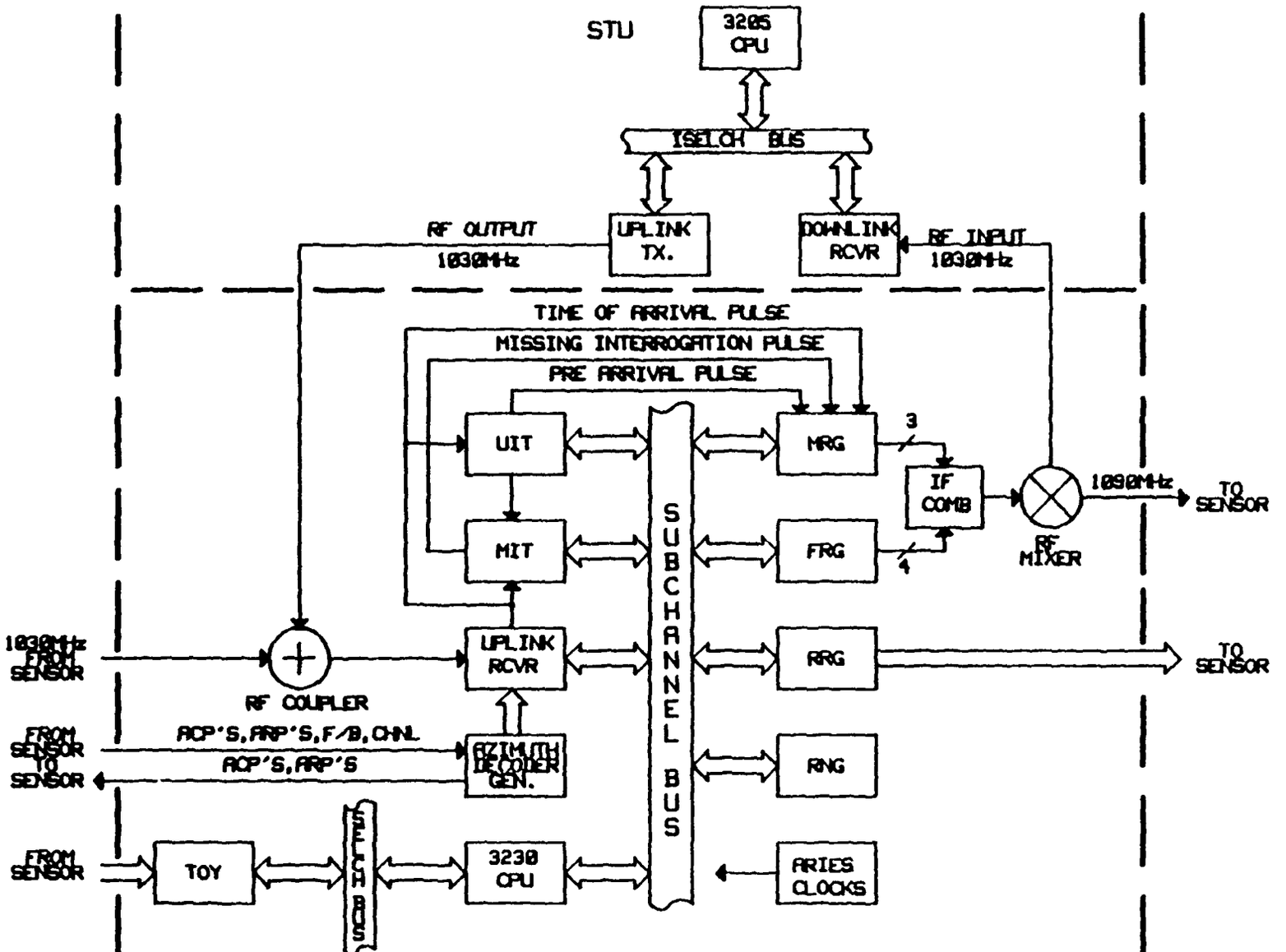


FIGURE 6-1. ARIES FUNCTIONAL BLOCK DIAGRAM

The MRG is interfaced to the computer through the Modeled Reply Controller (MRC) and to the Mode S sensor through three Modeled ARIES Target (MAT) reply generators. Likewise, the FRG is interface to the computer through a similar controller called the Fruit Reply Controller (FRC) and to the Mode S sensor through four Fruit ARIES Target (FAT) reply generators. Instead of obtaining replies from the computer, as the MRC does, the FRC obtains replies from two random process generators, the ATCRBS Random Process Generator (ARPG), and the Mode S Random Process Generator (MRPG). The Random Process Generators (RPG) generate ATCRBS and Mode S fruit, respectively, with random power, random offboresight angle, random ATCRBS code/Mode S data, and random values of exponentially distributed delay-to-trigger time. The overall fruit characteristics, such as fruit rate, mainbeam/sidelobe (M/S) ratio, and ratio of fixed to random ATCRBS code are controlled by the computer.

Outputs from the seven reply generators are combined in the IF combiner network to produce a single set of IF signals at 60 MHz. The set of IF signals are converted to RF in the IF to RF convertor network to provide a single set of RF inputs at 1030 MHz to the ARIES input ports of the Mode S sensor.

The TOY Receiver accepts TOY information from the Mode S sensor TOY clock. This information is fed to the computer for recording.

The Azimuth Decoder/Simulator accepts ACP and ARP signals from the Mode S antenna. These signals can be fed back to the sensor in a daisy-chain fashion when the sensor is operating with a real Mode S antenna. Simulated ACP and ARP signals are sent to the Mode S sensor when it is operating without an external azimuth source. The Azimuth Decoder/Simulator also transmits azimuth words to the Uplink Receiver.

Channel Status and Active Antenna signals are received from the Front End of the Mode S sensor. These signals determine the ARIES input/output channel to receive interrogations/transmit replies. The Channel Status signals also determines which input channel is selected to accept ACP and ARP signals and TOY information.

ARIES range timing is achieved with a single 16-MHz system clock. Range counters in the Uplink Receiver and in each of the MATs are reset each time an All-Call interrogation is decoded by the Uplink Receiver. This event is indicated by the ATCRBS/All-Call time-of-arrival (TOAR) pulse. These counters are not reset by Roll-Call interrogations. STU range timing is achieved with a separate 16-MHz clock. Range counters in the STU are reset in the same manner as the ARIES range counters.

The ARIES system is required to simulate targets with a minimum range of 1 nmi from the Mode S sensor. The round-trip delay to a 1-nmi target is about 12 microseconds (μs). Adding the 3-μs reply delay of the ATCRBS transponder, it is clear that only 15 μs are available for ATCRBS to respond to an All-Call interrogation when simulating a close-in ATCRBS target. Fifteen μs is insufficient time to provide for the interrupt latency of the computer. Therefore, it is necessary to predict the time-of-arrival of ATCRBS interrogations. The Universal Interval Timer (UIT) is used for this purpose. The computer transfers to the UIT the expected time of the next ATCRBS interrogation. During the interval between one All-Call interrogation and the next, the software has time to prepare the entire set of replies for the next predicted interrogation. Approximately 56 μs before the expected arrival time of the next ATCRBS interrogation, the UIT requests an interrupt, thereby indicating that no more Mode S interrogations are anticipated and that 56 μs later, the MAT's must be prepared to reply to an All-Call interrogation.

Other devices shown on figure 6-1, which are also interfaced to the computer, are the Radar Report Generator (RRG), the Missing Interrogation Timer (MIT), and the Random Number Generator (RNG). The RRG generates radar target reports in the ASR-9 format or the CD-2 format for serial transmission to the radar input port of the Mode S sensor. The MIT is used to indicate that an expected All-Call interrogation was not detected. The RNG provides the computer with a rapid succession of pseudo-random numbers.

Circuitry is included which is used to loop-test ARIES under computer control, completely independent of the Mode S sensor under test. This is provided by the STU which is capable of testing for:

a. Correctness of simulated reply data contents at the RF output.
b. Correctness of amplitude in the SUM, DIFF, and OMNI RF outputs.
c. Time of reply or range readout.
d. Azimuth of reply when transmitted.
e. Correctness of interrogation decoding at the RF input.

All of the FAA Technical Center built ARIES devices described in this manual are interfaced to the Concurrent computers using the standard Concurrent programmed I/O protocol. Thus, it is necessary to understand the pertinent Concurrent conventions in order to completely understand the operations of each piece of hardware described. These conventions are summarized below[1].

The ARIES devices, shown in figure 6-1, are either connected to the Subchannel Controller (SCC) private bus, or the Buffered Selector Channel (BSELCH) private bus as shown in figure 6-2. Both are extensions of the processor's Multiplexer (MUX) bus to provide increased fanout capability and for hardware isolation protection between the ARIES devices and the processor. Collectively, these buses are referred to as the I/O bus. All data transfers to/from the general purpose interface (GPI) support halfword (16 bits) transfers except the GPI for the MIT which supports only byte (8 bits) transfers.

The GPI supports communications with any device whose address is between 1 and 255. All buses are false true type, i.e., low-level active, high-level inactive.

In a typical case, a device will receive an 8-bit address byte, a 16-bit command halfword, or a 16-bit data halfword from the processor as shown in figure 6-3. Likewise, the device will send an 8-bit address byte, an 8-bit status byte, or a 16-bit data halfword to the processor. All transfers are over the same 16-bit bi-directional I/O bus. When only a byte of data is transferred, that byte is passed over the lower eight bus lines.

To communicate with a device, its associated interface must be addressed. Refer to figure 6-4 during the following description. When the device is addressed, an 8-bit address code is placed on the I/O bus lines D080 through D150. The GPI transceiver logic passes the address to the address decoder. The address is compared with the setting selected on the address dip switches. The switch setting defines the device code or address.

[1]For complete description of the Concurrent I/O protocols and conventions, refer to "Concurrent Universal Logic Interface Instruction Manual" Concurrent documentation No. M48-013.

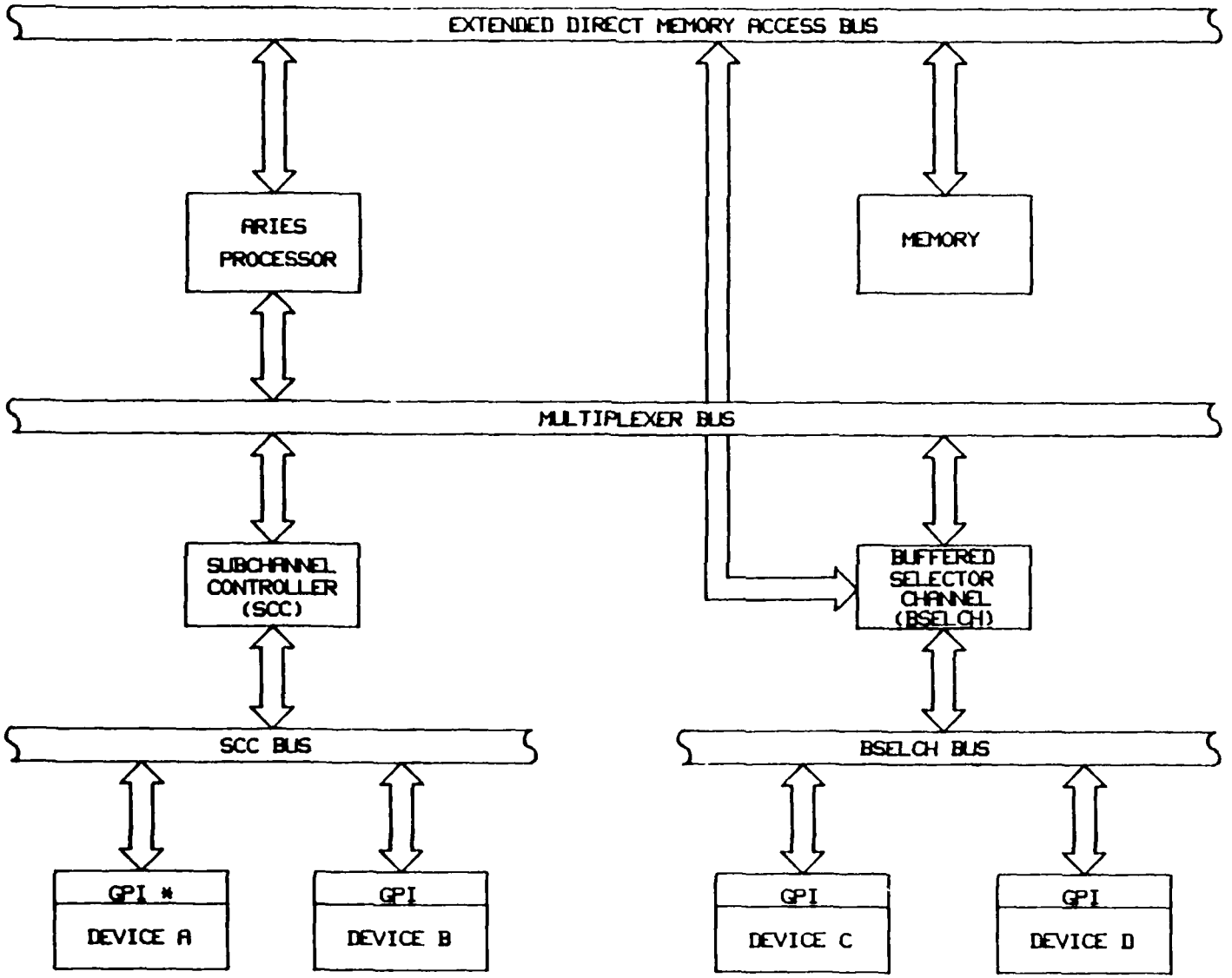


FIGURE 6-2. CONCURRENT STANDARD I/O BUS STRUCTURE

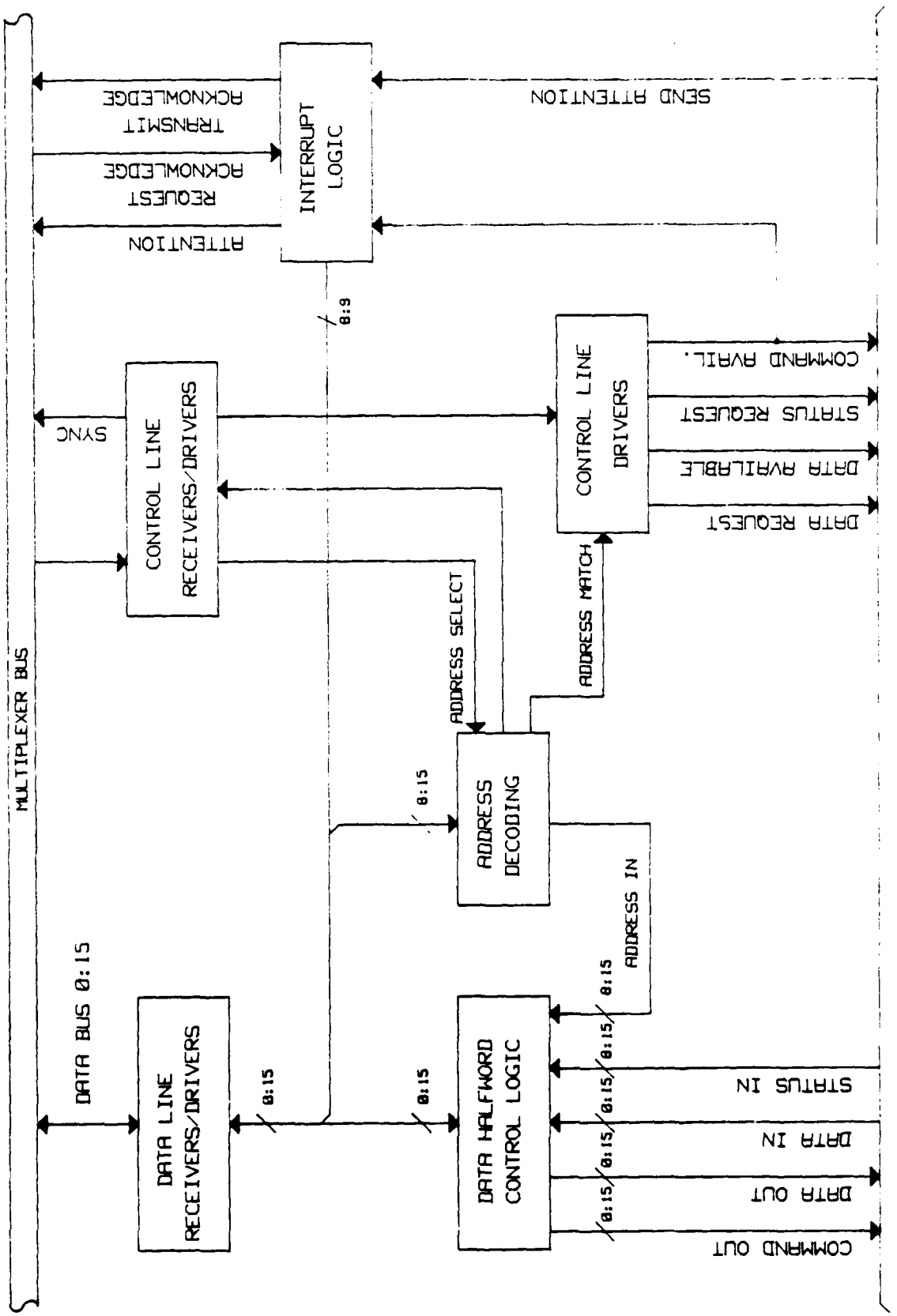


FIGURE 6-3. GENERAL PURPOSE INTERFACE BLOCK DIAGRAM

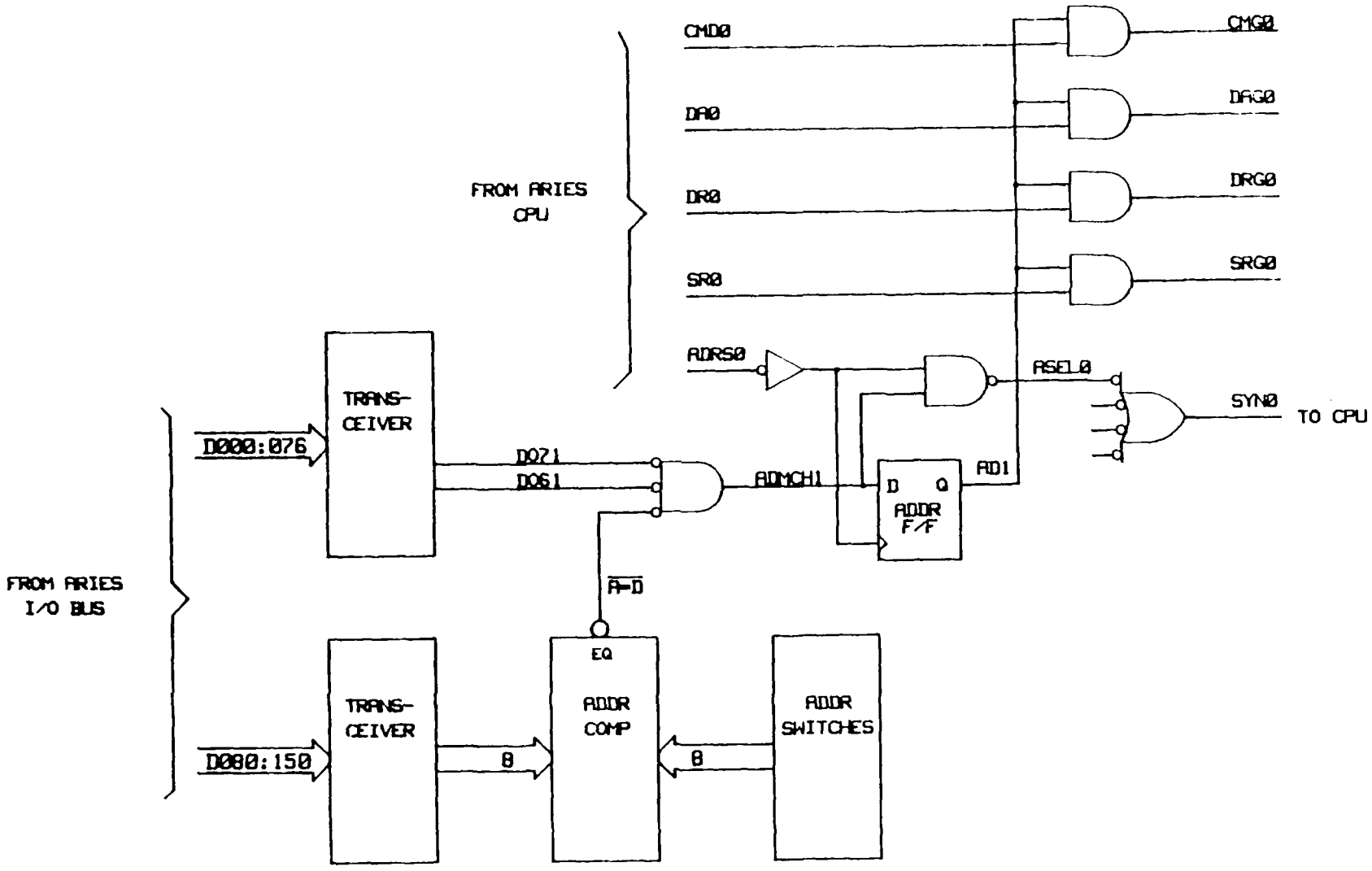


FIGURE 6-4. ADDRESS DECODER LOGIC

Upon receiving an address match the signal $\overline{A=D}$ goes active low. This output signal is ANDed with D061 and D071 to activate the address match line ADMCH1. The address control line ADRS0 then strobes the ADMCH1 line into the address (ADDR) flip-flop. Note that both D061 and D071 must be low (active) for the address match to occur.

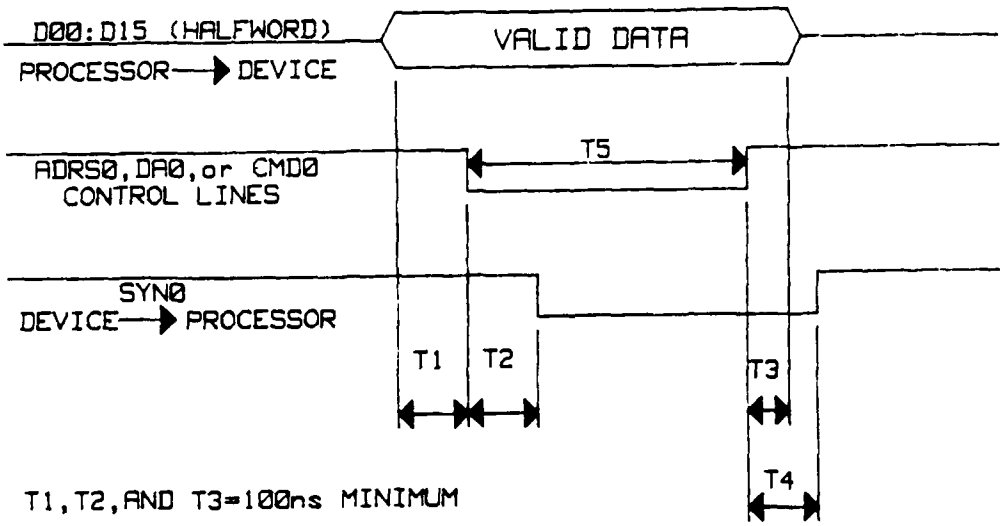
The synchronize signal SYN0 is returned to the processor, during the presence of ADRS0, via the address select ASEL0 line. Note that an OR gate is used here for returning the other device control sync lines, as well. The set output AD1 from the ADDR flip-flop activates all of the other I/O control lines to the device. When another device is addressed, the ADMCH1 line goes inactive low, causing the ADRS0 strobe line to reset the ADDR flip-flop, disabling the device. Thus, only one device may be addressed at any time. During the address cycle, only the device that was addressed returns a SYN0.

The device interrupt mode can be in one of three states; enabled, disabled, or disarmed. The interrupt mode is determined via command to a previously selected device. Bits D081 and D091 of the command are used to control the state of the GPI. Table 6-1 provides a quick outline of how this is accomplished. When the GPI is enabled, an interrupt can be queued in and sent to the computer. When the GPI is disarmed, an interrupt can be queued in but the interrupt is not sent to the processor until enabled. When the GPI is disarmed, an interrupt cannot be queued or sent to the computer. During a system power restore, the GPI interrupt mode is placed in the disarmed state.

TABLE 6-1. INTERRUPT COMMAND STRUCTURE

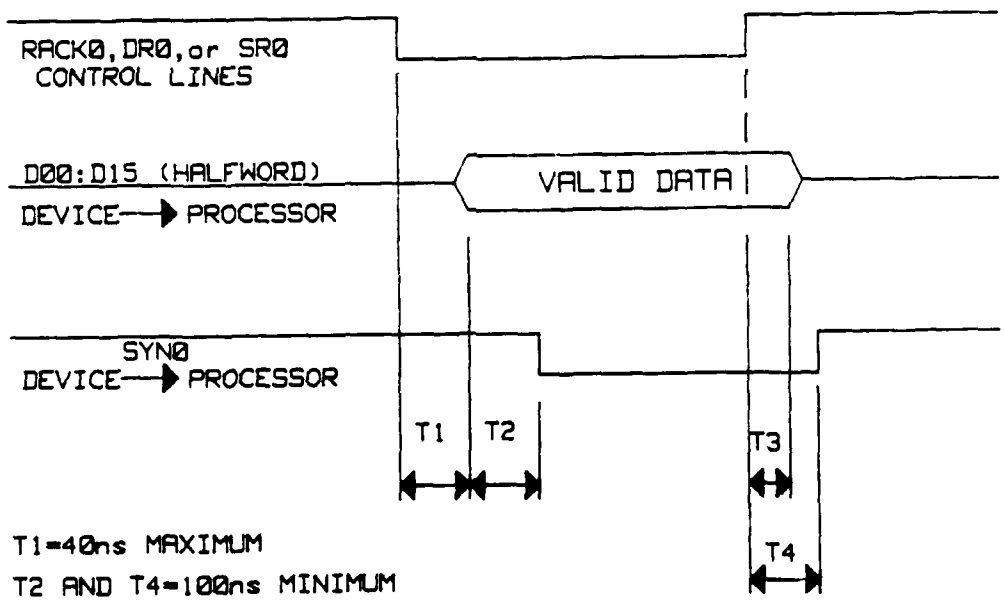
INTERRUPT STATE	COMMAND BIT 8	9
No Change	0	0
Enable	0	1
Disable	1	0
Disarm	1	1

Data and command outputs to the device, as well as addressing the device, are handled similarly. The timing waveforms for these output operations is shown in figure 6-5 (Output). The processor places data on the I/O bus then follows with the appropriate control line signal (Data Available (DA0), Command (CMD0), or Address (ADRS0)). This stagger (T_1) is guaranteed to be at least 100 nanoseconds (ns). The selected GPI then sends the SYN0 signal to the processor acknowledging that the control line was received. The responds turn around time (T_2) is guaranteed to be at least 100 ns after receiving the control line. The processor terminates the control line when the GPI has acknowledged and then removes the data lines. The control line/data line removal time (T_3) is guaranteed to be at least 100 ns. Typically, the control line signal remains active (T_5) for 450 ns. The SYN0 test line signal is removed once the control line signal is removed, completing the output operation cycle. This stagger (T_4) is guaranteed to be at most 130 ns. It is the removal of the control line signal that is used, in almost all cases, to latch in the data on the I/O bus.



T1, T2, AND T3=100ns MINIMUM
T4=130ns MAXIMUM
T5=450ns TYPICAL

OUTPUT



T1=40ns MAXIMUM
T2 AND T4=100ns MINIMUM
T3=130ns MAXIMUM

INPUT

FIGURE 6-5. MULTIPLEXER CHANNEL TIMING

Data and status inputs from the device, as well as requesting the address of a device, are handled similarly. The timing waveforms for these input operations is shown in figure 6-5 (Input). The processor activates the desired control line signal (Data Request (DR0), Status Request (SR0), or Request Acknowledge (RACK0)). The selected GPI places the requested data on the I/O bus, then sends the SYN0 test line signal to the processor acknowledging the request. The control line/data line respond time (T_1) is guaranteed to be at most 40 ns. The stagger time (T_2) between the request and acknowledge is guaranteed to be at least 100 ns. The processor removes the control line signal when the SYN0 is received with a minimum delay (T_4) of 100 ns. Since SYN0 is generated by the control lines being coupled through gates, it is in turn removed with a maximum delay (T_3) of 130 ns, completing the input operation cycle.

6.1 ARIES SYSTEM CLOCK.

The ARIES digital system operates synchronously using two basic clocks derived from a single 16.0-MHz oscillator (the STU digital subsystem operates using its own 16.0-MHz oscillator). This clock shares board space with the UIT, the MIT, and the RNG on the multiple device board located in slot No. 13 of the ARIES digital chassis. In its normal or "run" mode, the ARIES clock logic generates continuous 30-ns pulses spaced 62.5 ns apart (16 MHz), and 62.5-ns pulses spaced 250 ns apart (4 MHz). In its diagnostic or "single step" mode, the ARIES clock logic generates a single 30-ns pulse on the 16-MHz system clock line (16-MHz single step switch depressed), or a group of four 30 ns pulses on the 16-MHz system clock lines and a single 62.5-ns pulse on the 4-MHz system clock lines (4-MHz single step switch depressed).

Operation of the ARIES clock may be visualized by examining the block diagram in figure 6.1-1. The sinusoidal output of the 16-MHz crystal oscillator is shaped to a symmetrical rectangular pulse output. In the "run" mode, the 2:1 multiplexer passes its symmetrical input to flip-flop IC_1 which generates the asymmetrical 30 ns 16-MHz clock output. Every fourth pulse of the symmetrical 16 MHz waveform is also passed to flip-flop IC_2 to create the 62.5-ns wide 4.0-MHz output.

In the "single step" mode, one or four negative going pulses are selected by the "16-MHz single step" or the "4-MHz single step" switches, respectively, to drive the output D flip-flops IC_1 and IC_2.

6.2 UPLINK RECEIVER.

The Uplink Receiver consists of both analog and digital circuitry. The analog receiver is mounted on the STU RF Transmitter/Uplink Receiver panel located in slot No. 9 of the ARIES analog chassis. The digital receiver shares board space with the Azimuth Decoder/Simulator on the Uplink Receiver board located in slot No. 11 of the ARIES digital chassis. The functions of the analog circuitry is discussed in section 6.2.1 and the functions of the digital circuitry is discussed in section 6.2.2.

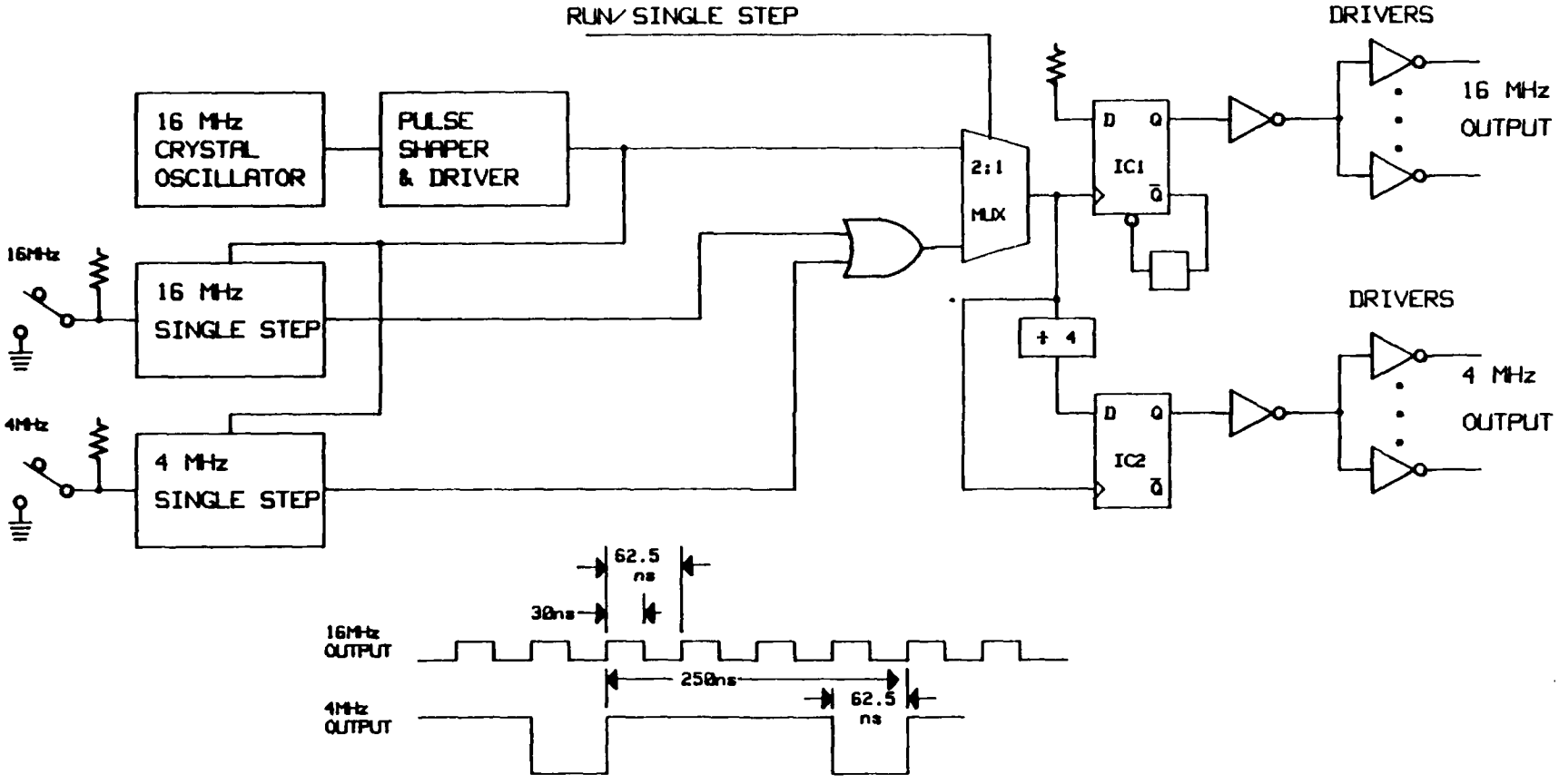


FIGURE 6.1-1. ARIES SYSTEM CLOCK BLOCK DIAGRAM AND WAVEFORMS

6.2.1 Analog Circuitry.

The input of the ARIES Uplink Receiver is coupled from the RF port of the Mode S sensor transmitter at 1030 MHz. The interrogation receiver is designed to receive signal levels within -10 to -1 decibels above 1 milliwatt (dBm). As shown in figure 6.2.1-1, the Mode S interrogations are coupled through a 20-decibel (dB) directional coupler then filtered through a 1030-MHz preselector. The directional coupler also serves as an input port to receive interrogations generated by the STU. The preselector eliminates undesirable, out-of-band interference (image, spurious, etc.) before being fed to the RF mixer. The RF mixer modulates the 1030-MHz signal with the local oscillator (LO) signal of 970 MHz to produce a 60-MHz IF signal. The 60-MHz signal is detected and amplified by the log amplifier which provides two output signals: (1) a log video signal, and (2) a conditioned 60-MHz signal. The log video signal is demodulated by the threshold circuit generating the pulse amplitude modulation (PAM) demodulation signals compatible to standard transister-to-transister logic (TTL). The conditioned 60-MHz signal is decoded by the differential phase shift keying (DPSK) demodulator into a TTL compatible output signal that contains the Mode S phase modulation information. Both of these signals are sent to the digital circuitry of the ARIES Uplink Receiver.

6.2.2 Digital Circuitry.

The Uplink Receiver digital circuitry is similar to that included in the Mode S transponder receiver. The block diagram of the receiver is shown in figure 6.2.2-1. It consists of the following logic modules; channel selection, mode decoding, data block assembly, and data transfer. Included in the data block assembly module is a 20-bit range counter and a Mode S 24-bit parity decoder. These modules, along with other support modules, will be discussed in detail in the following subsections.

6.2.2.1 Channel Selection.

ARIES operates in either the single or the dual antenna mode. In the single antenna mode, the ARIES accepts input from one of two RF channels. An RF channel consists of one interrogation input and three reply output (SUM, DIFF, and OMNI) lines associated with the input line. In the dual antenna mode, the ARIES switches between each RF channel. The mode in which the ARIES operates is dependent on the state of the Channel Active and the Antenna Select control lines received from the Front Ends of the Mode S sensor. These control lines also select the TOY channel and the azimuth input channel.

Tables 6.2.2.1-1 and 6.2.2.1-2 show the channel selection based on the state of the control lines. Figure 6.2.2.1-1 shows the logic that performs this selection. Channel A is selected by default if none of the control lines are active. If both Channel Active lines are on, the ARIES is assumed to be in the dual antenna mode and the RF channel selected is determined by the Antenna Select control lines. Under this configuration, the TOY data and the external azimuth data is received on channel A.

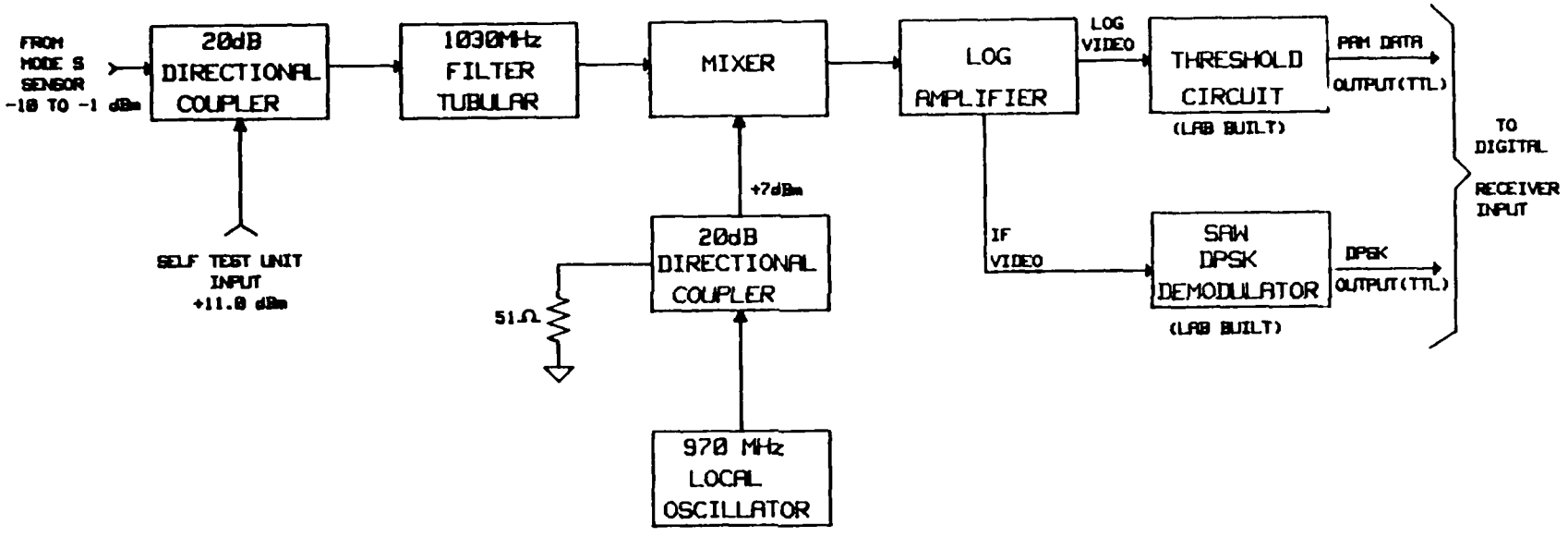


FIGURE 6.2.1-1. UPLINK RECEIVER ANALOG BLOCK DIAGRAM

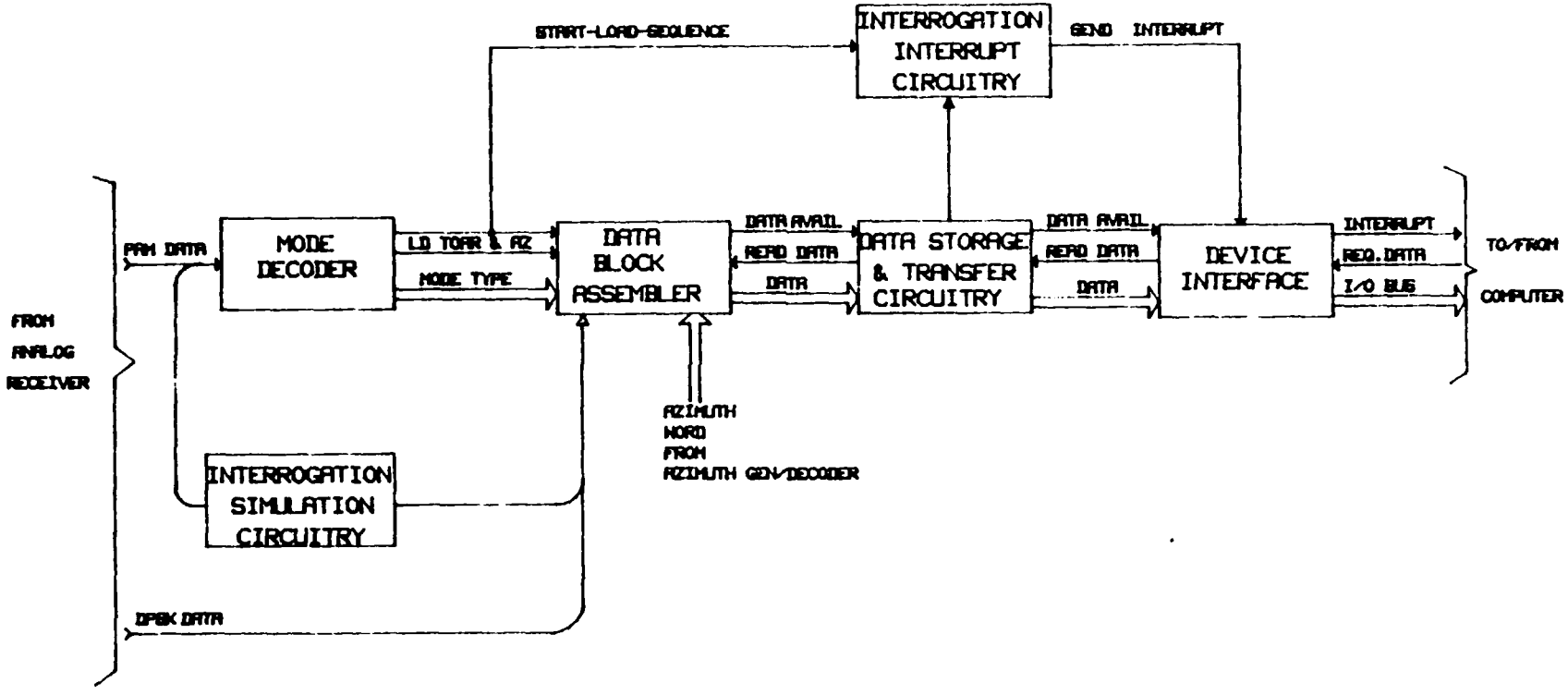


FIGURE 6.2.2-1. UPLINK RECEIVER DIGITAL BLOCK DIAGRAM

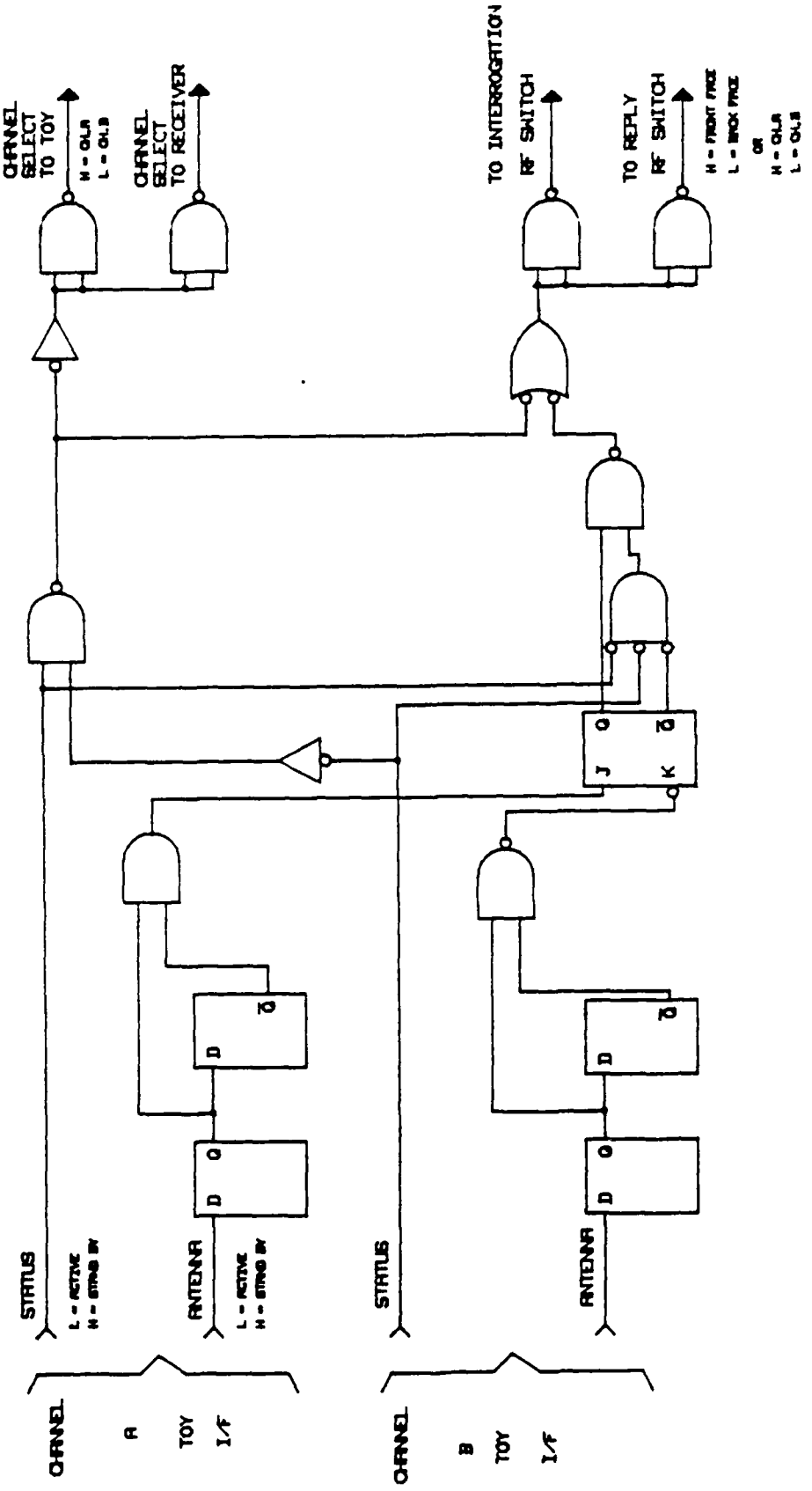


FIGURE 6.2.2.1-1. CHANNEL SELECT LOGIC

TABLE 6.2.2.1-1. TERMINAL (SINGLE ANTENNA) CONFIGURATION

Mode S Sensor				ARIES			
Front End No. 1		Front End No. 2		Channel		Antenna	
Channel	Antenna	Channel	Antenna	A	B	A	B
X	X	1	X	0	1	0	1
1	X	0	X	1	0	1	0

TABLE 6.2.2.1-2. EN ROUTE (DUAL ANTENNA) CONFIGURATION

Mode S Sensor				ARIES			
Front End No. 1		Front End No. 2		Channel		Antenna	
Channel	Antenna	Channel	Antenna	A	B	A	B
0	0	0	X	0	1	0	1
0	1	0	1	0	1	0	1
0	1	0	0	0	1	1	0

NOTES: (1) All lines are considered active low
(2) X denotes don't care state

6.2.2.2 Mode Decoding.

The mode decoding logic determines the mode of each interrogation received. This is done by testing the PAM input for specific pulse spacings and pulse widths. The block diagram in figure 6.2.2.2-1 shows how this is accomplished. The PAM input is fed through a 22-μs digital delay element constructed from a series of shift registers. The outputs from the delay element are tapped at 2, 5, 8, and 21 μs corresponding to the separation between the P_1 and P_2 (2 μs) pulses for the Mode S preambles and between the P_1 and P_3 (5, 8, 21 μs) pulses for the ATCRBS brackets as shown in figure 6.2.2.2-2. If a mode is detected by a NAND gate, the corresponding J/K flip-flop will be set.

If the interrogation detected is an ATCRBS type, the existence of a P4 pulse and its width are determined. This is done by starting the pulse detection sequencer at the time an ATCRBS bracket is detected. The sequencer generates a detection window over the time of an expected P4 lead-edge pulse and a detection window over the time of an expected wide P4 trail-edge pulse. If the P4 lead-edge pulse is detected by the appropriate NAND gate, the corresponding J/K flip-flop is set. Also, if the wide P4 trail-edge pulse is detected by the appropriate NAND gate, the corresponding J/K flip-flop is set. The timing diagram shown in figure 6.2.2.2-3 illustrates the detection sequence of a Mode A/Mode S All-Call interrogation.

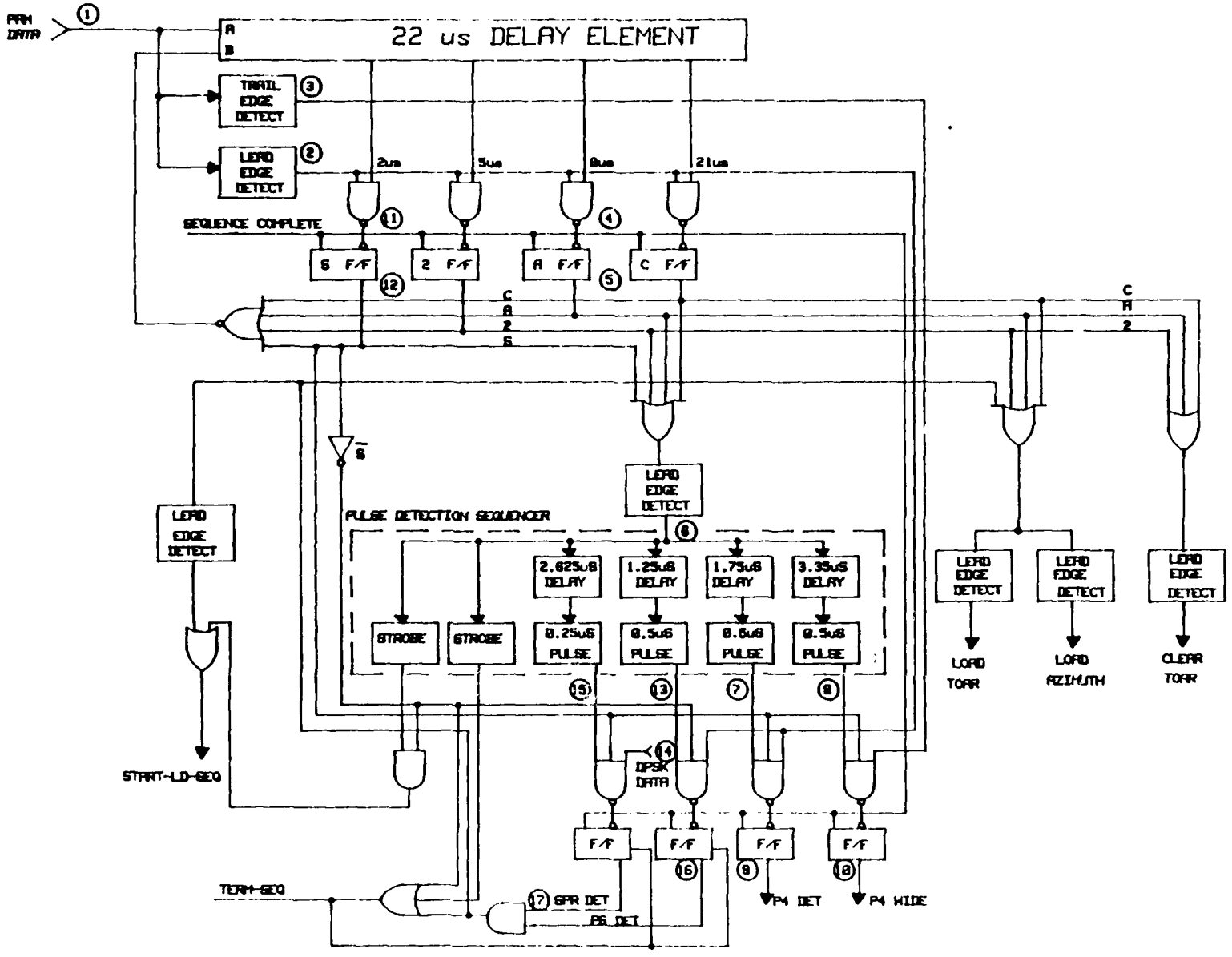
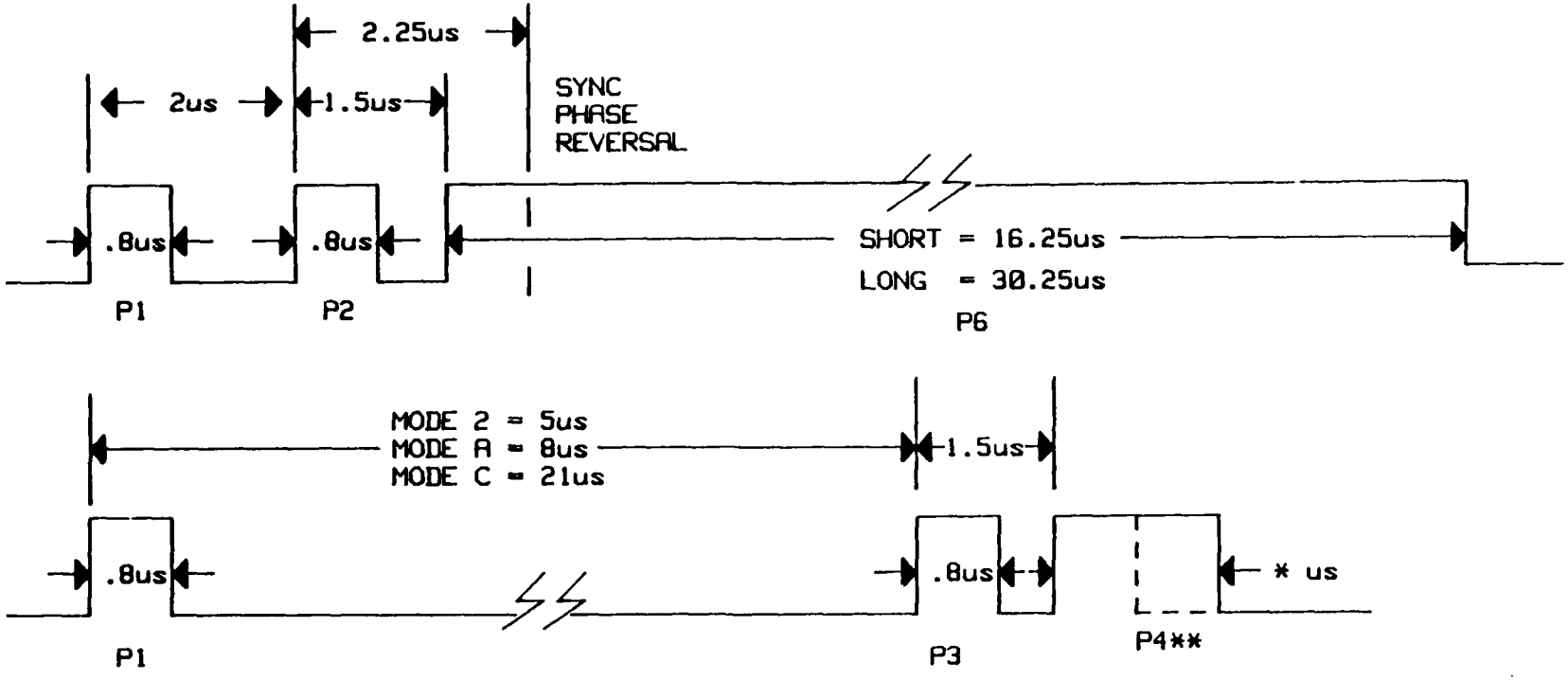


FIGURE 6.2.2.2-1. MODE DECODER BLOCK DIAGRAM



* ATCRBS/MODE S ALL-CALL = 1.6us
  ATCRBS-ONLY ALL-CALL = 0.8us
** P4 is used only for MODE A and MODE C

FIGURE 6.2.2.2-2. MODE DECODER WAVEFORM

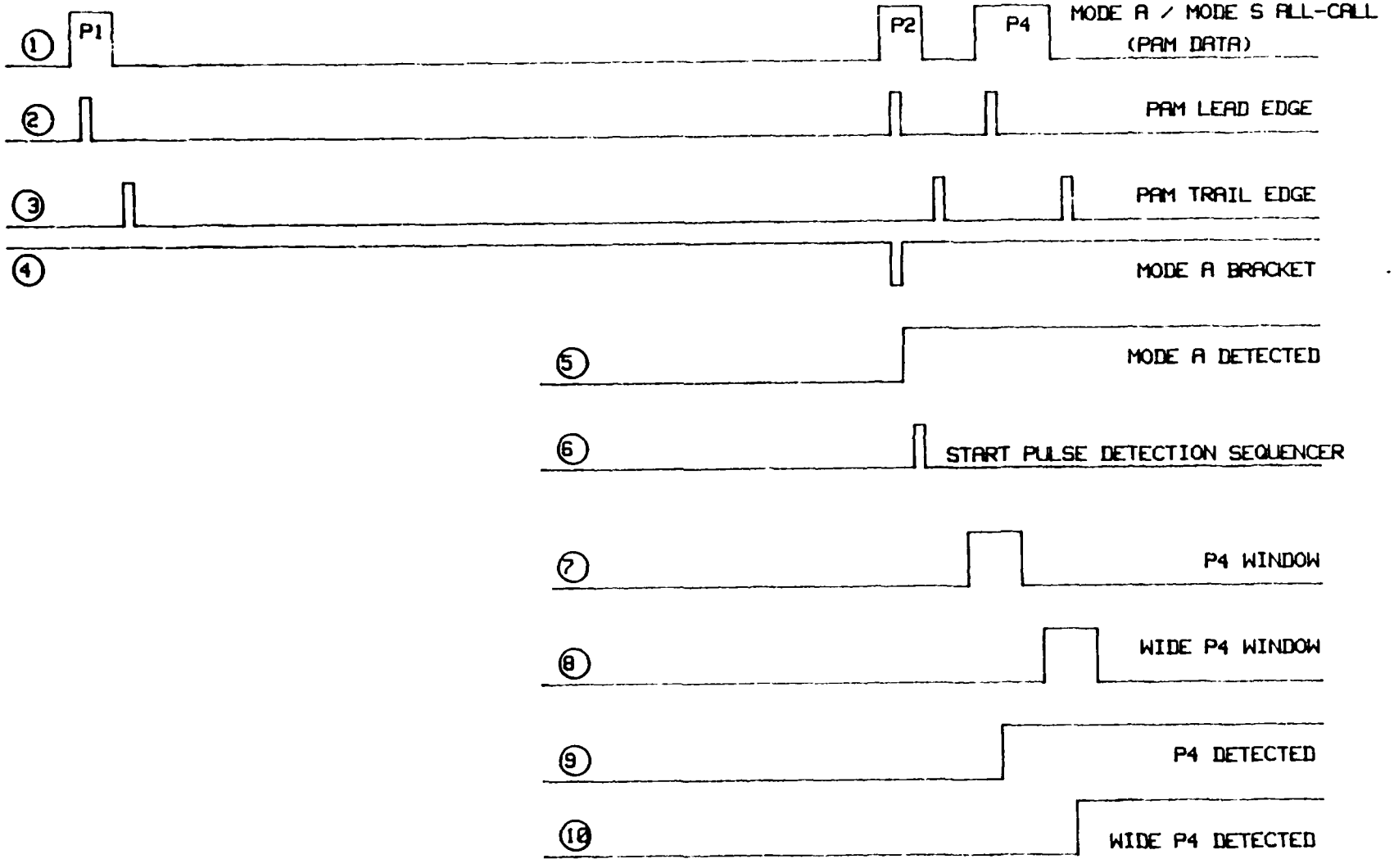


FIGURE 6.2.2.2-3. MODE A/ALL-CALL MODE DECODING TIMING DIAGRAM

Note the number shown for each signal corresponds to the numbered logic points shown in figure 6.2.2.2-2. In the case of mode 2 interrogations, there is no defined P4 pulse; the pulse detection sequencer is still triggered but no P4 detection windows are generated.

If the interrogation detected is a Mode S type, the existence of the P6 pulse and the alignment of the synchronous phase reversal (SPR) pulse is determined. This is done by starting the pulse detection sequencer at the point the preamble is detected. It, in turn, generates a detection window around the time of the expected P6 lead-edge pulse and a detection window around the expected DPSK data SPR followed by a test strobe to determine if these detections were successful. If the P6 lead-edge pulse and the SPR pulse are detected by their appropriate NAND gates, the corresponding J/K flip-flops are set. If either of these flip-flips are not set, the test strobe is passed through OR gate A to generate signal TERM-SEQ0. This signal resets the mode decoder logic and generates a receiver error interrupt flagging that a Mode S interrogation was possibly lost. The timing diagram shown in figure 6.2.2.2-4 illustrates the detection sequence of a Mode S Roll-Call interrogation. Note the number shown for each signal corresponds to the numbered logic points shown in figure 6.2.2.2-2.

Upon receipt of each interrogation, the mode decoding logic requests an interrupt to inform the computer of the arrival of the interrogation. The mode decoding logic samples the time of arrival and the antenna boresight azimuth at the time an ATCRBS bracket or the Mode S SPR are detected. A Start-Load-Sequence pulse is generated to start the interrogation data block assembly logic collecting information on the interrogation.

6.2.2.3 Interrogation Data Block Assembly.

A block diagram of the interrogation data block assembly logic is shown in figure 6.2.2.3-1. Note that this logic consists of:

a. 20-bit TOAR Counter with a 20-bit TOAR Register
b. 14-bit Azimuth Register
c. Interrogation Type Encoder with a Type Register
d. Mode S Decoder with a 16-bit Data Register
e. 16-bit by 16-word Output Buffer (First In First Out Memory (FIFO))
f. Sequence Controllers No. 1 and No. 2

The TOAR counter runs at a 16-MHz rate which corresponds to one range unit (RU) per clock cycle. The TOAR counter is sampled each time an interrogation is received. (The TOAR counter is sampled at the time an ATCRBS bracket or a Mode S SPR is detected.) The time data is latched into the TOAR Register by the mode decoding logic. Upon the receipt of an All-Call interrogation, the range counter is reset by the mode decoding logic after it is sampled.

The azimuth data is latched into the Azimuth Register at the same time the TOAR counter is sampled. This data is taken from the Azimuth Decoder/Simulator (see section 6.3).

The Type Encoder encodes the interrogation type based on the settings of the mode decoder flip-flops and, if the interrogation is a Mode S Roll-Call, the first two data bits of the message (the two most significant bits (MSB's) of the UF field). The encoded type data is then latched into the Type Register. The encoding is performed according to table 6.2.2.3-1.

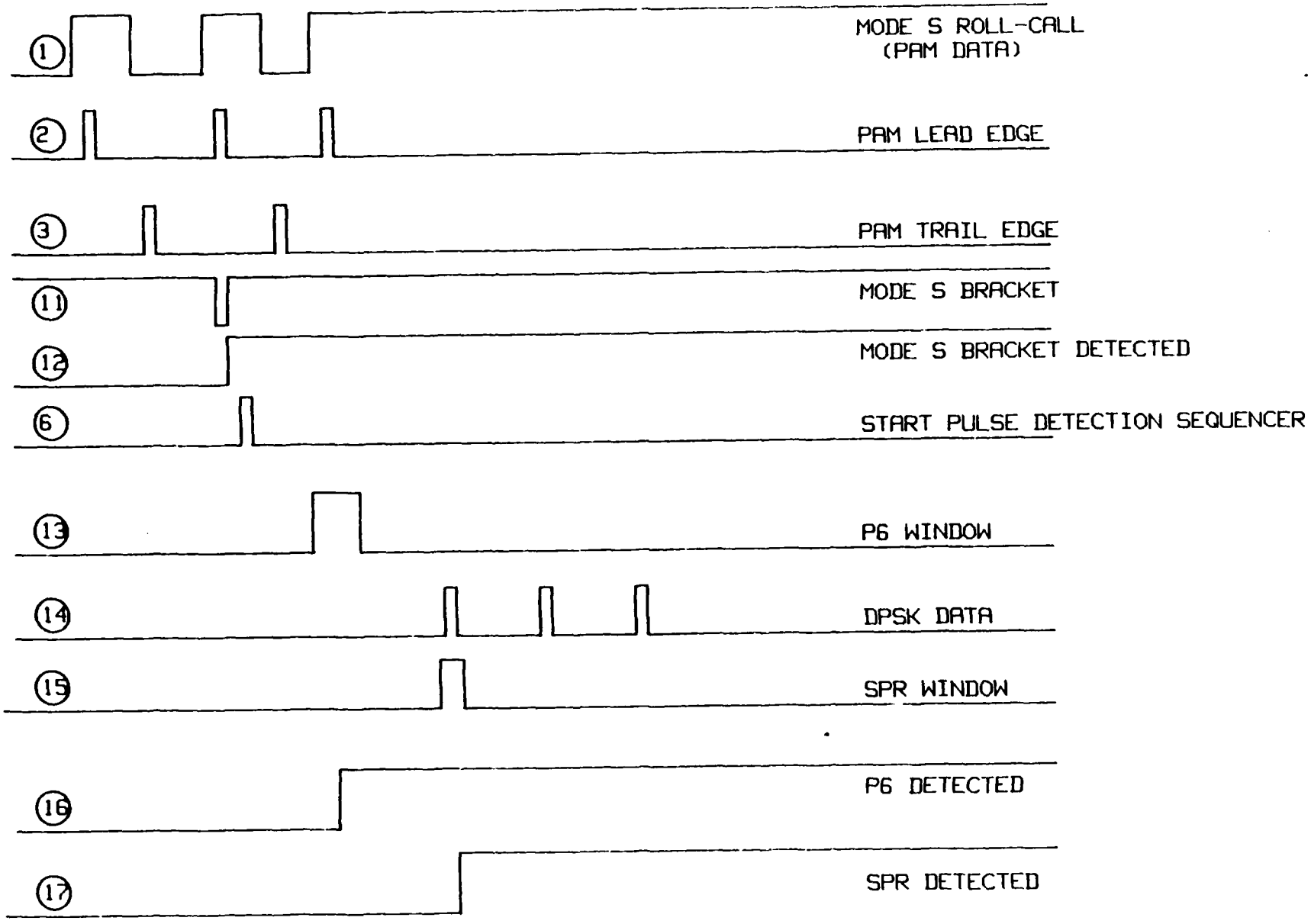


FIGURE 6.2.2.2-4. MODE S ROLL-CALL MODE DECODER TIMING DIAGRAM

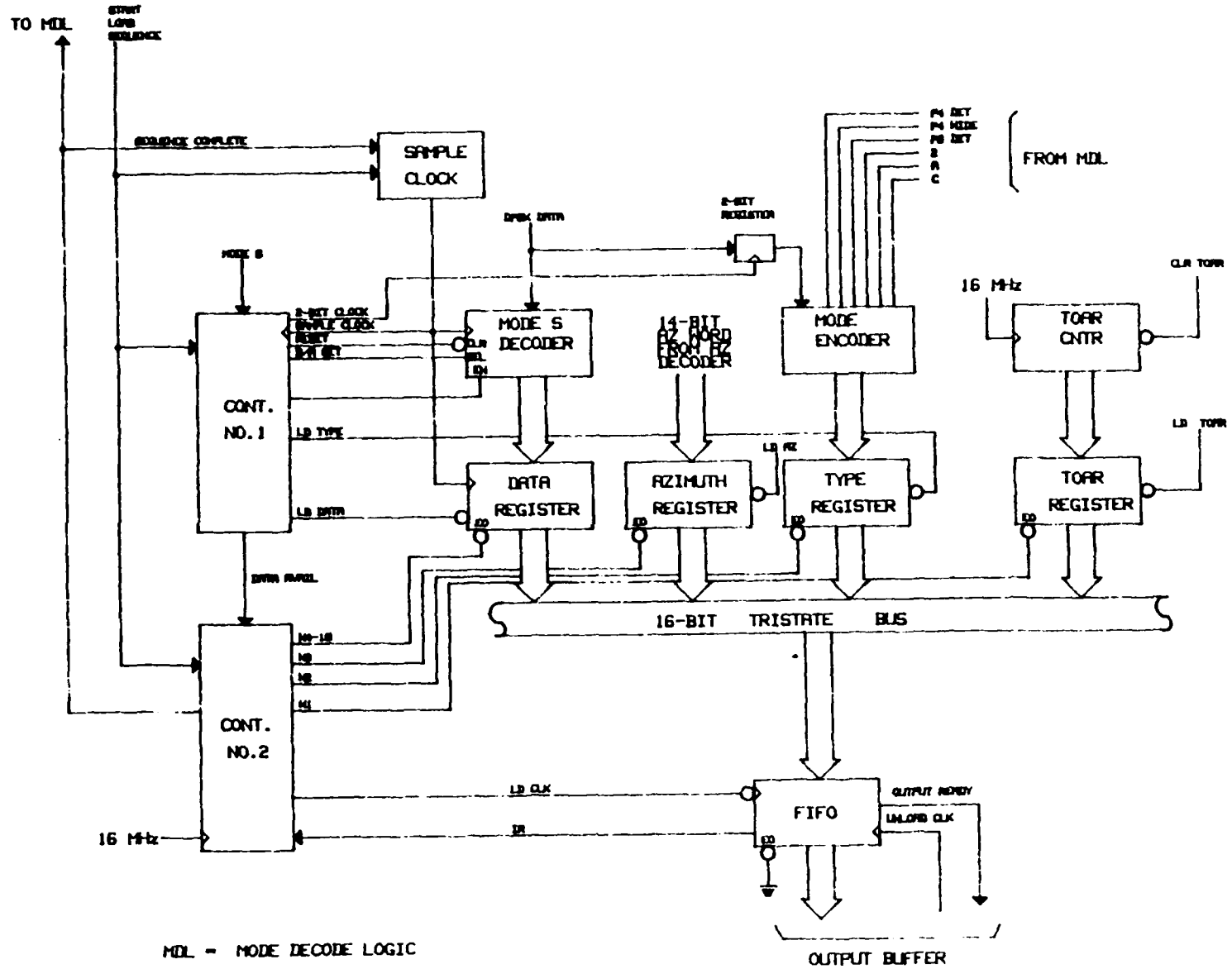


FIGURE 6.2.2.3-1. DATA ASSEMBLY BLOCK DIAGRAM

TABLE 6.2.2.3-1. INTERROGATION MODE ENCODER ROM

UF5	UF4	P6 DET	P4 WD	P4 DET	C	A	2	ROM ADDR.	ENCODE DATA	COMMENTS
A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	(HEX)	(HEX)	
0	0	0	0	0	0	0	1	01	1	Mode 2
0	0	0	0	0	0	1	0	02	2	Mode A
0	0	0	0	1	0	1	0	0A	3	Mode A Only All-Call
0	0	0	1	1	0	1	0	1A	4	Mode A All-Call
0	0	0	0	0	1	0	0	04	5	Mode C
0	0	0	0	1	1	0	0	0C	6	Mode C Only All-Call
0	0	0	1	1	1	0	0	1C	7	Mode C All Call
0	0	1	0	0	0	0	0	20	8	Mode S Short
1	0	1	0	0	0	0	0	A0	9	Mode S Long (Comm A)
1	1	1	0	0	0	0	0	E0	A	Mode S Long (Comm C)
All remaining addresses									0	Improper mode decode

The interrogation data is assembled into a 10-word data block by the two sequence controllers. The process is initiated when the mode decoding logic sends the *Start-Load-Sequence* pulse to both controllers. This pulse starts the 4-MHz sample clock (SAMPLE-CLK) which clocks Controller No. 1 and the Mode S Decoder logic. Controller No. 2 automatically transfers the upper 16-bits of the TOAR Register into the message assembler FIFO and then waits for additional data, signaled by the DATA-AVAIL strobe from Controller No. 1. When the first DATA-AVAIL strobe is received, the interrogation mode has been encoded and latched in the TYPE Register. At this time, Controller No. 2 transfers the contents of the TYPE Register and the remaining four least significant bits (LSB's) of the TOAR Register into the FIFO, then transfers the azimuth word into the FIFO forming the second and third words of the interrogation data block. Upon receipt of the next seven DATA-AVAIL strobes, the contents of the Data Register are transferred into the FIFO forming the fourth through the tenth words of the interrogation data block. Once the tenth word is transferred, Controller No. 2 generates the sequence complete (SEQ-COMP0) pulse which is fed back to the mode decoding logic, clearing all of the mode J/K flip-flops, resetting Controller No. 1, and stopping the SAMPLE-CLK logic. Controller No. 2's sequence is independent of the type of interrogation received.

Controller No. 1 executes one of three possible load sequences depending first on whether the mode decoder Mode S J/K flip-flop is set or not. If the Mode S J/K flip-flop is not set, an ATCRBS load sequence will be executed. Since there is no data associated with an ATCRBS/All-Call, the sequence is a simple one. First the Data Register is loaded with "0." On the following sample clock, the mode-encoded data is latched into the TYPE Register and the DATA-AVAIL pulse is sent to Controller No. 2. Then a set of DATA-AVAIL pulses are generated, one each 0.5 μs, 0.75 μs after the first pulse getting Controller No. 2 to read the Data Register and pack the remaining part of the message data block with "0."

If the Mode S J/K flip-flop is set, one of two Mode S sequences is selected depending on whether the interrogation is short or long. This is accomplished by looking at the MSB of the UF field. When this bit is low, the short Mode S sequence is executed. When this bit is high, the long Mode S sequence is executed.

In the case of Mode S interrogations, the Mode S decoder assembles the data bits and removes the error detection encoding from the Mode S address field. This is done by the parity decoding logic shown in figure 6.2.2.3-2. The parity decoder consists of a 24-bit shift register and a sum modulo-2 network connected to the shift register at appropriate points corresponding to the 24^{o} polynomial generator. (For more detail, see "Mode S National Standard," FAA Order 6365.1A.) Both the Mode S address and the parity are decoded. The shift register is first cleared. The received DPSK data is then fed into the shift register. After 32 shifts (Mode S surveillance) or 88 shifts (Comm A or Comm C), the shift register will contain 24 parity bits. These 24 parity bits are then summed bit by bit, modulo-2, with the remaining 24 bits from the input DPSK data stream to produce the Mode S address.

When the SPR is detected, the sample clock will be activated and begin to sample the DPSK data at 0.25 μs intervals. As the DPSK data is being fed into the parity decoder, it is also being fed to a 16-bit shift register without modification via the multiplexer. The contents of the shift register is parallel loaded into the Data Register after each 16 samples so that the entire 56-bit or 112-bit data block can be transferred to the output buffer. After the first two data bits are sampled, the type and length of the Mode S interrogation is determined. The first bit allows Controller No. 1 to select the multiplexer at the appropriate time, so that the last 24 bits may be sent to the 16-bit shift register from the parity decoder. The last 24 bits are the Mode S address.

6.2.2.4 Data Transfer Sequence.

Since it is not possible to anticipate the time of arrival of Mode S interrogations, it is crucial that the received data is transferred to the computer upon receipt of a Mode S interrogation in the minimum possible time. This is accomplished by initiating I/O data transfer immediately following the detection of a valid Mode S interrogation even before all of the data has been received. The identical data transfer technique is used for ATCRBS interrogations. Direct memory access data transfer is too inefficient for short data block transfers, such as 10-word blocks; therefore, this approach was not selected.

A block diagram of the Data Transfer Sequence logic is shown in figure 6.2.2.4-1. The output of the FIFO, the 16-bit by 1024-word memory buffer, and the device interface share a common tristate bus. The Output Control logic monitors the output ready of the FIFO to move data into the memory buffer. When data is available, the control logic passes the write address counter value to the address lines of the memory buffer and places the output of the FIFO on the tristate bus. The data is then transferred into the buffer and the write address counter is incremented to the next address.

If loading and reading the FIFO occur at the same time, the internal timing of the FIFO can be effected preventing data from falling through the internal registers of the FIFO correctly. Therefore, the simultaneous read and write operations had to be avoided. Loading the data into the FIFO had to be performed in real time but reading data from the FIFO could be delayed to prevent the above problem. This is accomplished by having Controller No. 2 generate a four-clock wide pulse, referred

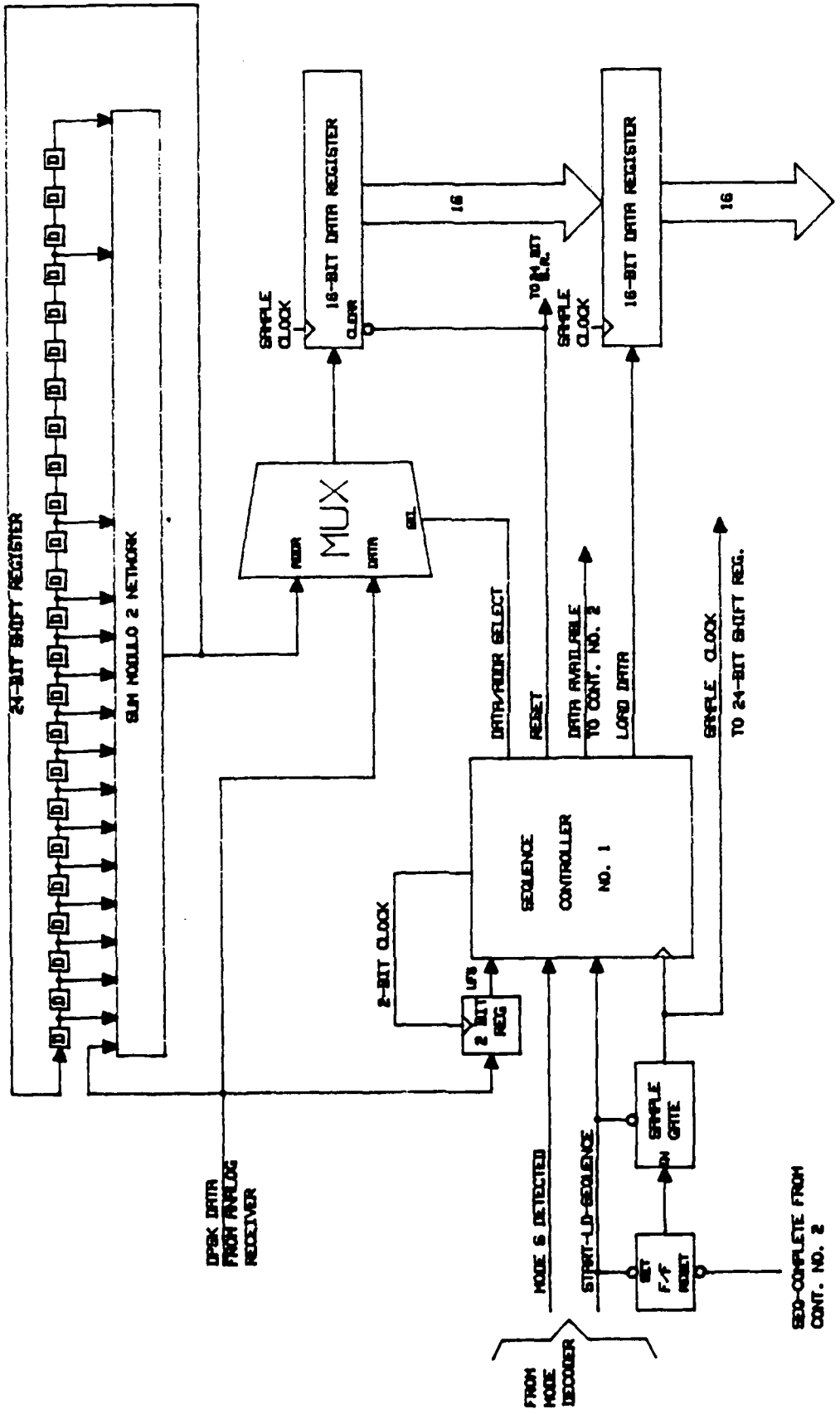


FIGURE 6.2.2.3-2. MODE S PARITY DECODING BLOCK DIAGRAM

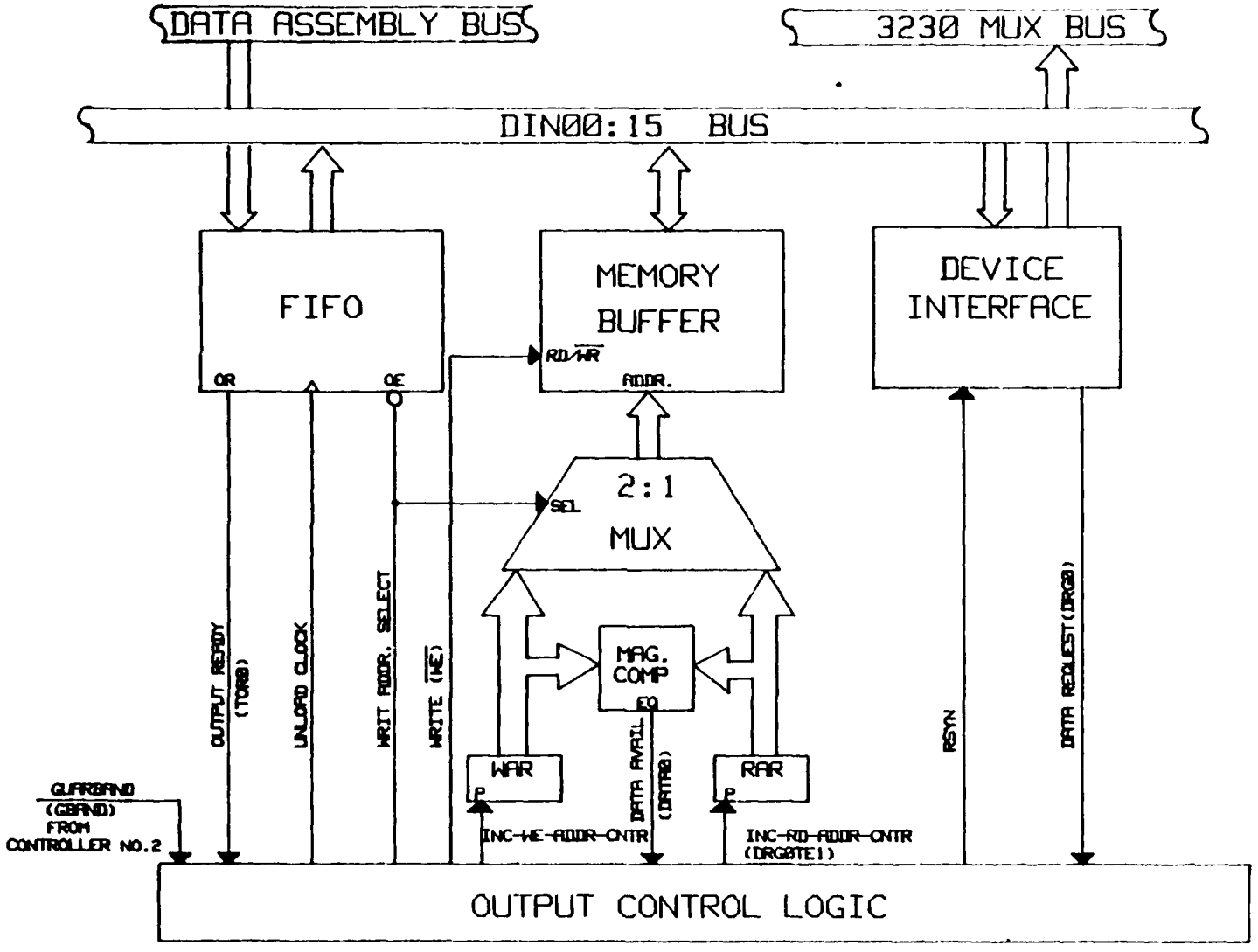


FIGURE 6.2.2.4-1. OUTPUT TRANSFER SEQUENCE BLOCK DIAGRAM

to as the Guardband, three clocks before it generates the load FIFO clock. If the output control logic tries to generate the unload clock (UNLD-CLK1) when the Guardband is present, the UNLD-CLK1 will be delayed for four clocks (250 ns delay) before it is generated. The logic diagram of how this is done is shown in figure 6.2.2.4-2.

Upon receipt of each interrogation, an interrupt is generated to inform the computer of its arrival. This is done by the mode decoding logic as described previously. When an interrogation interrupt is received, the computer fetches 10 words from the Uplink Receiver.

For the case of an ATCRBS interrogation, the complete interrogation data block is loaded into the memory buffer before the computer begins to request data. So no contention for the use of the tristate bus occurs between loading and unloading the memory buffer. A timing diagram of the data transfer sequence for an ATCRBS interrogation is shown in figure 6.2.2.4-3a.

For the case of a short Mode S interrogation, the interrogation data block is still being loaded into the memory buffer when the computer begins to request data. This is due to the fact that it takes an additional 15.0 μs, after the detection of the interrogation, for all of the Mode S data to be received. So contention for the bus exist in getting data into and out of the memory buffer. The output control logic that handles the bus contention is shown in figure 6.2.2.4-4.

Referring to figure 6.2.2.4-4, assume that data becomes available, signal TOR0 goes low, before a data request is received. The output of NOR gate A goes high forcing the output of NOR gate B low. The output of gate A is then synchronized to the 16-MHz system clock and the necessary control signals are generated to transfer a word from the FIFO to the memory buffer. If during this interval a data request is issued, i.e., DRG0 goes low, the output of gate B remains low until the signal TOR0 is removed. Once the signal TOR0 is removed, the output of gate A goes low enabling gate B to go high, which in turn, forces gate A from being turned on until the data request is completed. At this time signal RSYN0 is sent back to the computer acknowledging the data request was received. The computer reads the data then removes the data request signal. This in turn generates the signal DRG0TE1 that increments the memory buffer read address counter to the next location. A typical timing diagram of the data transfer sequence for a short Mode S interrogation is shown in figure 6.2.2.4-3b.

For the case of long Mode S interrogation, the interrogation data block is still being loaded into the memory buffer when the computer begins to request data and at a certain point the computer attempts to read data before it is loaded into the memory buffer. This is due to the fact that it takes an additional 29.0 μs, after the detection of the interrogation, for all of the long Mode S data to be received. So not only does bus contention have to be addressed but also whether the data had been received at the time the computer requests it. This problem is handled by the third input signal of NOR gate B (DATA0) shown in figure 6.2.2.4-4.

The signal DATA0 is the equal output of the 12-bit magnitude comparator (refer to figure 6.2.2.4-1). The inputs to the comparator are the outputs of the write address counter and the read address counter of the memory buffer. When both address counters match, no data exists in the memory buffer and the signal DATA0 goes high. This forces the output of NOR gate B low essentially pausing any data request until data becomes available. Also, NOR gate A is enabled giving priority to transferring data from the FIFO into the memory buffer. As soon as the next

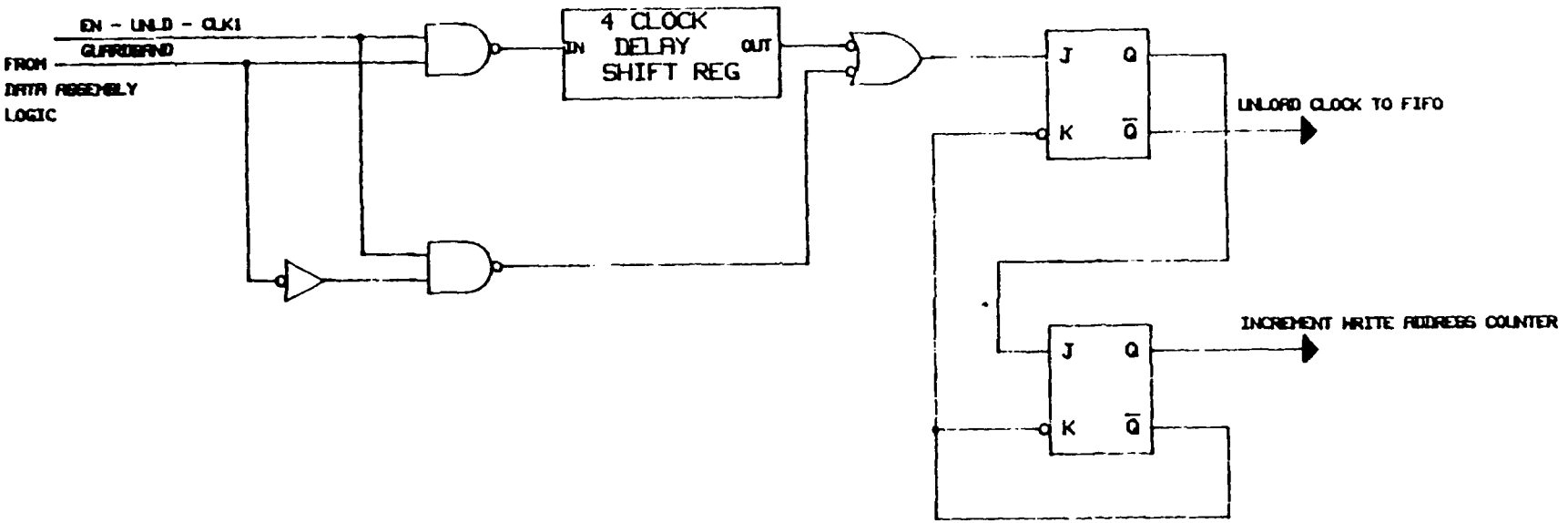
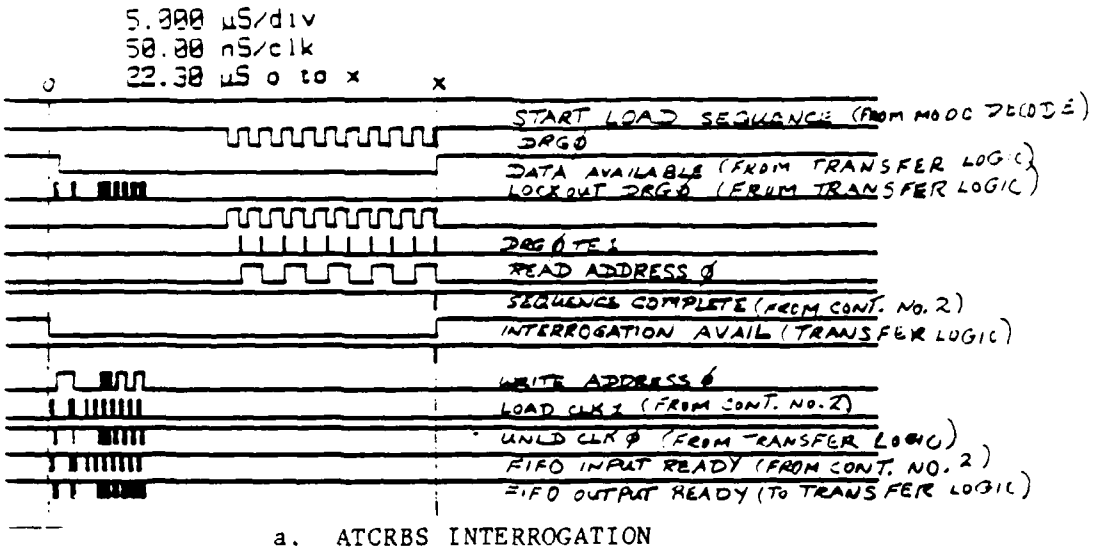
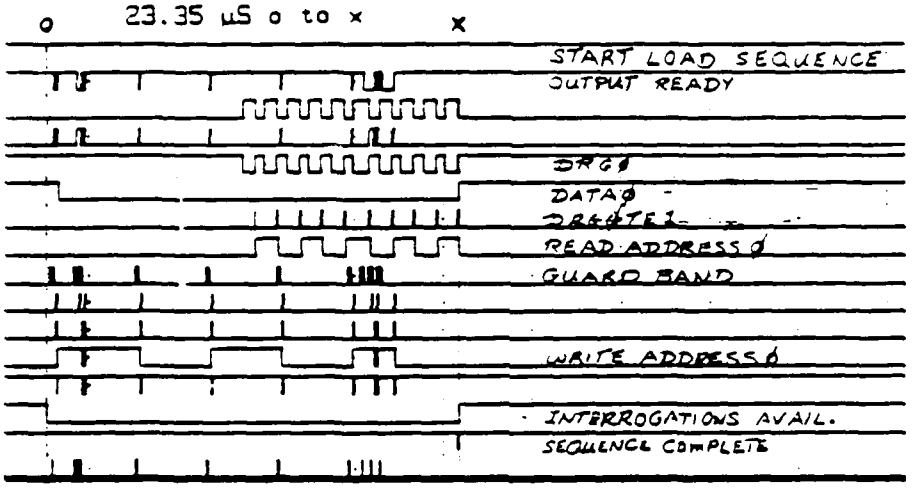


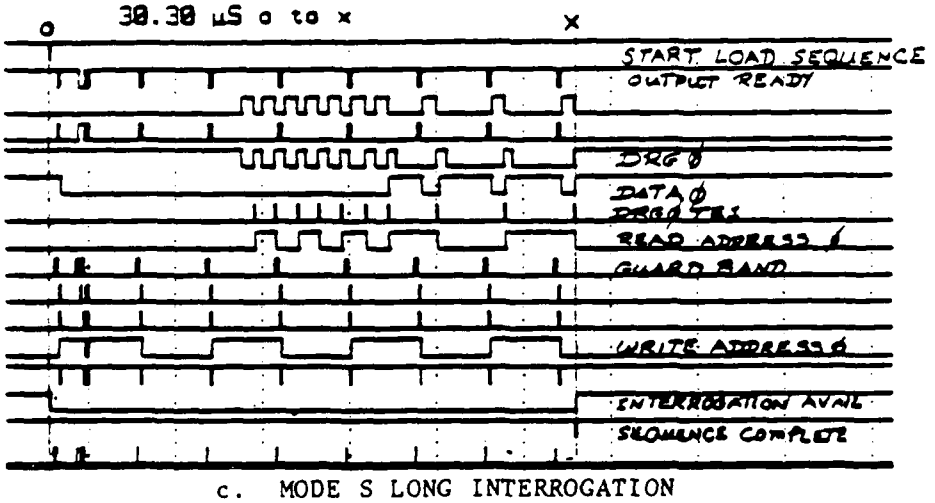
FIGURE 6.2.2.4-2. FIFO UNLOAD CLOCK LOGIC



a. ATCRBS INTERROGATION



b. MODE S SHORT INTERROGATION



c. MODE S LONG INTERROGATION

FIGURE 6.2.2.4-3. UPLINK RECEIVER TIMING WAVEFORM DIAGRAM

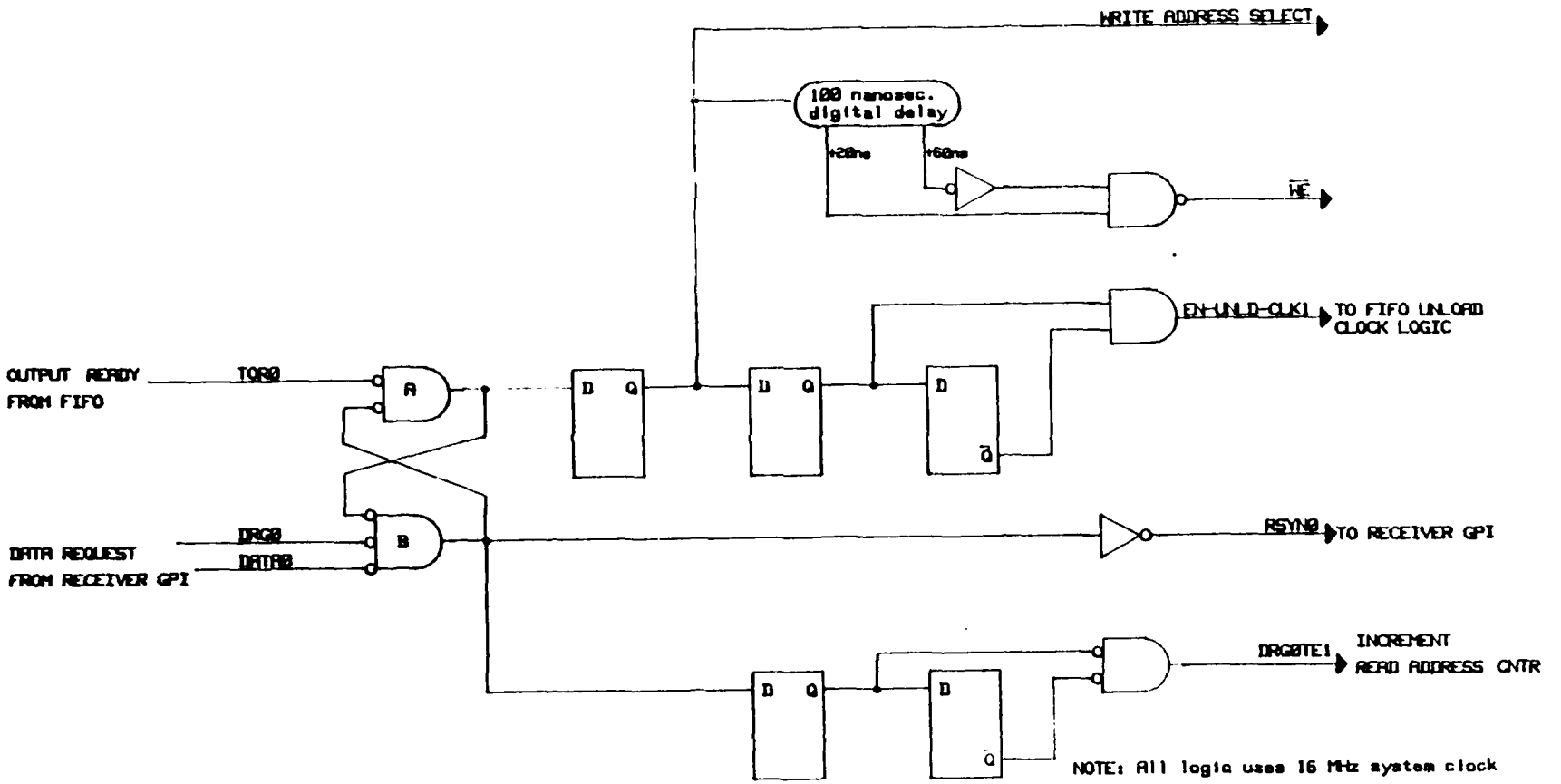


FIGURE 6.2.2.4-4. BUS CONTENTION CONTROL LOGIC

word is loaded into memory, the write address counter is incremented by one and no longer matches the read address counter. The signal DATA0 then goes low enabling gate B to pass the signal DRG0 through. A timing diagram of the data transfer sequence for a long Mode S interrogation is shown in figure 6.2.2.4-3c.

6.2.2.5 Interrogation Interrupt Request Circuitry.

Since temporary buffering of interrogation data block messages is necessary in the Uplink Receiver, a means of notifying the processor that these data blocks existed is also necessary. This is accomplished by keeping track of the number of interrogation interrupt requests to be serviced. This is the purpose of the interrogation interrupt request circuitry shown in figure 6.2.2.5-1.

Assuming that this logic has just been initialized, the interrogation interrupt counter contains a value of 0. This value fed to the A inputs of the 8-bit magnitude comparator matches the fixed 0 value placed on the B inputs. This generates a high signal on the equal output of the comparator selecting the A input of the 2:1 multiplexer M. When an interrogation is received, the mode decoding logic generates the Start-Load-Sequence pulse which passes through the multiplexer to produce the request attention pulse, RATN0. On the following clock cycle, the interrogation interrupt counter is incremented by one and the equal output of the comparator goes low selecting the B input of the multiplexer for the next interrupt request.

Ten data requests are executed each time an interrogation interrupt service routine (ISR) is executed. The 10 data requests correspond to the complete interrogation data block transferred to the computer. Counter $CNTR_1$ keeps track of the number of words read. After 10 words are read, Counter $CNTR_1$ decrements the interrogation interrupt counter $CNTR_2$ by one and one clock later sends a pulse to the B input of the multiplexer. If the existing interrogation interrupt counter is decremented back to 0, the A input of the multiplexer is selected preventing the pulse at the B input from passing and generating a false interrupt request.

Now assuming that a second interrupt request is generated before the first interrupt is completed, the second Start-Load-Sequence pulse is prevented from passing through the multiplexer. The interrogation interrupt counter is incremented by 1, i.e., in this case the counter will now contain a value of 2. As soon as 10 words are read by the computer, the interrogation interrupt counter is decremented by 1, i.e., in this case, the counter will now contain a value of 1. Since the value of the counter is not 0, the pulse generated by $CNTR_1$ is passed through the multiplexer to generate the second interrogation interrupt.

In the event that completion of one interrupt request occurs simultaneously with the initiation of another interrupt request, the interrogation interrupt counter value remains unchanged. This is accomplished by feeding the decrement and increment lines to the EXCLUSIVE NOR gate C, then feeding the output to the enable input of the counter. If both signals to the gate are active at the same time, no enable pulse is generated leaving the value of the counter unchanged.

The interrogation interrupt counter is cleared prior to each expected All-Call interrogation. This is accomplished by using the 56 EARLY pulse generated by the UIT to reset the counter. This is performed to purge any Roll-Call interrogation data blocks that may still be remaining in the receiver. The reason for this action is a result of how the operational software anticipates when an All-Call interrogation will occur. When the 56 EARLY pulse is generated, an UIT interrupt

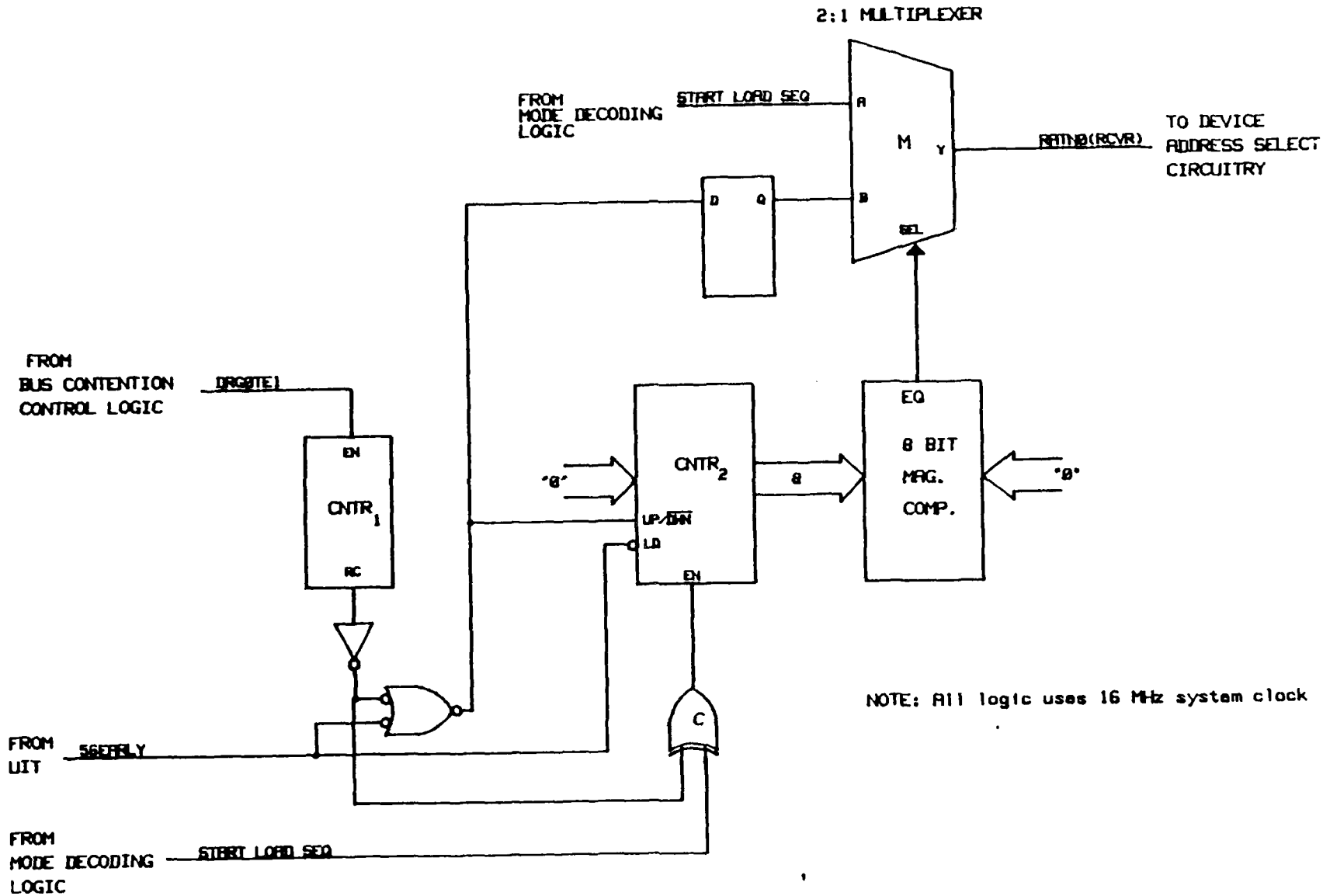


FIGURE 6.2.2.5-1. INTERROGATION INTERRUPT REQUEST CIRCUITRY

is generated signaling the operational software to prepare for an All-Call interrogation processing cycle. (See section 5.1.5 of the ARIES Software Principles of Operation Manual for further details.) If one or more Roll-Call interrogations still remain in memory, illegal interrogation errors would be generated by the operational software since only an All-Call is expected. Also, there would be insufficient time to process any more Roll-Call interrogation interrupts and respond with an appropriate reply before the next All-Call interrogation.

6.2.2.6 Device Interface Interrupt Modification.

The device interrupt circuitry for the Uplink Receiver is a modification of the Concurrent standard described in section 6. The device address is modified to indicate two different interrupt conditions. For one condition, the Uplink Receiver generates an interrupt each time an interrogation is received. The operational software will then execute one of two interrogation processing routines; the ATCRBS interrogation processing (AIP) routine or the Mode S interrogation processing (MIP) routine. When the UIT interrupt is received, the operating software prepares to execute the AIP routine on the next receiver interrupt. The MIP routine is executed on subsequent interrupts until the next UIT interrupt is received. So any interrupt generated by the Uplink Receiver would be treated as a request to process an incoming interrogation.

However, the Uplink Receiver can generate an interrupt if an error condition is detected, i.e., the terminate Mode S sequence mentioned in section 6.2.2.1 would cause an error interrupt to be generated. To identify the error condition, the device address is modified causing the operational software to treat the receiver as a completely different device. This is accomplished using the logic shown in figure 6.2.2.6-1 which sets the LSB of the device address high if an error interrupt request is issued.

When the pulse Start-Load-Sequence is generated by the mode decoding logic, it is first sent to the interrupt request circuitry. This logic then generates an active low pulse $RATN0_{(RCVR)}$ which sets J/K flip-flop IC_1 on the following clock cycle. At this moment both inputs of NAND gate A are high. The output of gate A is fed back to NAND gate B, disabling it, and to NOR gate C, enabling it. The output of gate A is then lead-edge detected to produce the pulse SATN0 on the output of AND gate E. This pulse in turn sets the device attention (ATTN) flip-flop.

The computer responds with a request acknowledge (RACK0 goes low) and reads the device address when the SYN0 line goes active low. Since NAND gate B is disabled, its output is high, causing the LSB of the device address SW151 to be 0. After the device address is read, RACK0 goes high causing the ATSYN0 signal to be removed. One clock later, the ATSYN0 signal is trail-edge detected and fed to NOR gates C and D. Since gate C is enabled, the pulse is fed to J/K flip-flop IC_1 resetting it on the next clock cycle.

If an error is detected and the device interface interrupt logic is not in the disarmed mode, that is if NAND gate F is enabled, the examine bit EX1 is lead-edge detected to produce the active low pulse $RATN0_{(STAT)}$. One clock later, J/K flip-flop IC_2 is set placing a high on one input of NAND gate B. If an interrogation interrupt request is still active NAND gate B remains disabled and the error interrupt request is in a pending mode. If no interrogation interrupt request exist or if it is completed, gate B is enabled. The output of gate B is

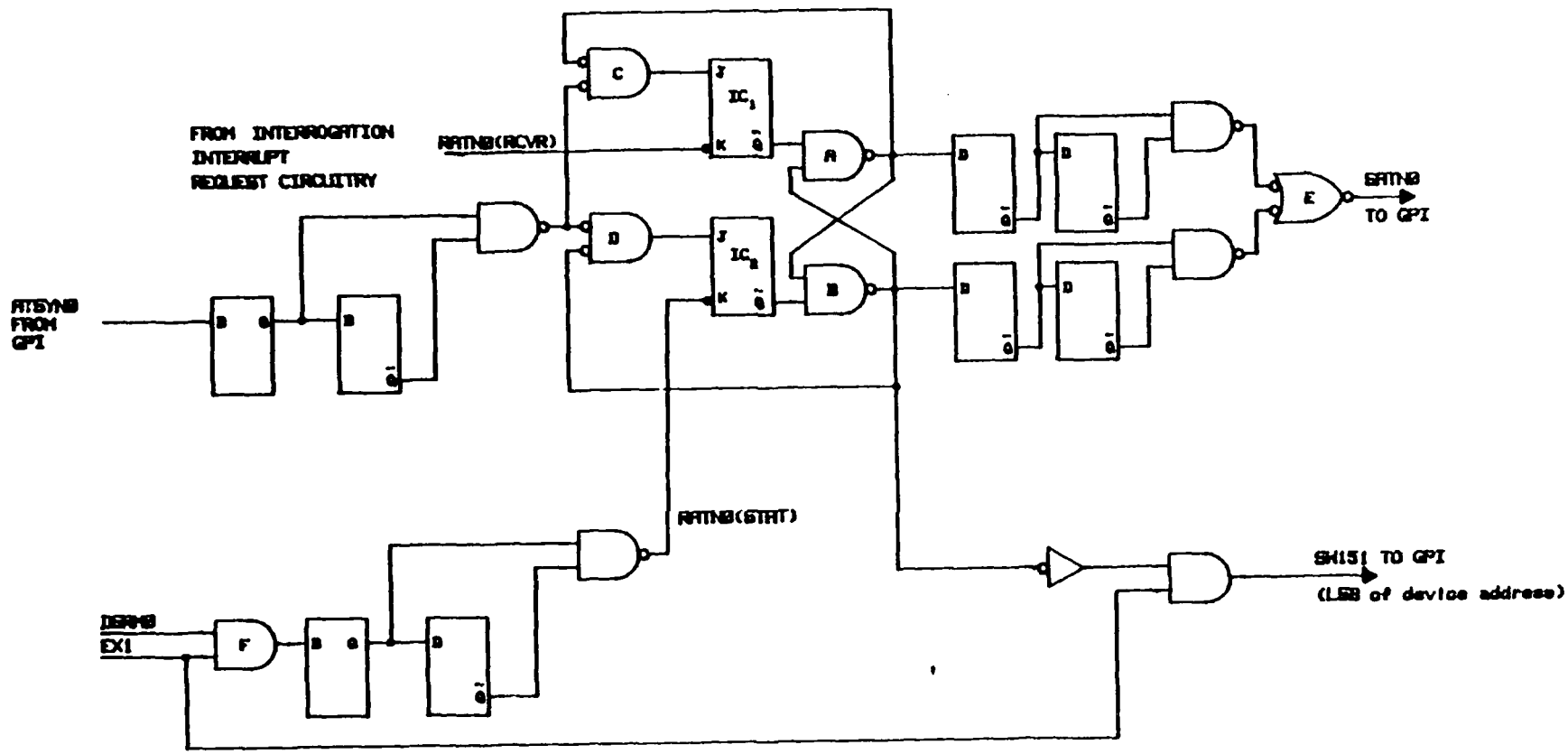


FIGURE 6.2.2.6-1. MODIFIED DEVICE ADDRESS SELECT CIRCUITRY

fed back to NAND gate A, disabling it, and to NOR gate D enabling it. Also the LSB, of the device address, is set to 1. Therefore, the device address for an error interrupt is one unit higher then the standard device address for an interrogation interrupt. On the following clock cycle, the pulse SATN0 is placed on the output of AND gate E setting the ATTN flip-flop.

Similarly, when the device address is read, the ATSYN0 signal is removed, generating the active low pulse. This pulse is fed to J/K flip-flop IC_2 since NOR gate D is enabled causing the flip-flop to be reset on the following clock cycle.

6.2.2.7 Interrogation Simulation Circuitry.

The interrogation simulation circuitry was integrated into the receiver to assist in testing the operational readiness of the Uplink Receiver. This logic injects one of eight predefined simulated interrogations into the digital front end of the receiver. The interrogation may be generated once or repeatedly. The eight interrogations listed below were selected to exercise as many of the receiver's decoding capabilities:

a. Mode 2
b. Mode A Only All-Call
c. Mode C/Mode S All-Call
d. Mode A/Mode S All-Call
e. Mode S Short (Altitude)
f. Mode S Short (Identification)
g. Mode S Long (Altitude)
h. Mode S Long (Identification)

A block diagram of the interrogation simulation circuitry is shown in figure 6.2.2.7-1. Predefined interrogations are stored in an 8-bit by 512-word read only memory (ROM). The upper four output lines contain the PAM data streams while the lower four output lines contain the corresponding DPSK data streams. The ROM is partitioned into the lower 256 words and the upper 256 words to produce the 8 PAM and 8 DPSK outputs needed. These outputs are fed to their respective 4:1 multiplexer and the desired pair (PAM and DPSK data stream) is selected via the device command. Both data streams are fed through clocked flip-flops to filter out transient pulses generated at the output of the ROM as its address is updated. The data streams are then coupled into the receiver.

A special 4-MHz logic clock (4 MHz PCLK) is used to clock the ROM address counter. This clock is generated by dividing the 16-MHz system clock by four. The purpose of this clock is to allow simulated interrogations to be injected into the receiver during different phases of the 4-MHz system clock.

When the single shot pulse (TRIG0) is generated, J/K flip-flop IC_1 is set enabling the ROM address counter. The counter begins at 0 and increments at a rate of 4 MHz until its reaches its maximum count. At which time, the ripple carry is generated. On the following clock lead edge, flip-flop IC_1 is reset and the counter is disabled with a value of 0. In this mode, the selected interrogation is generated once.

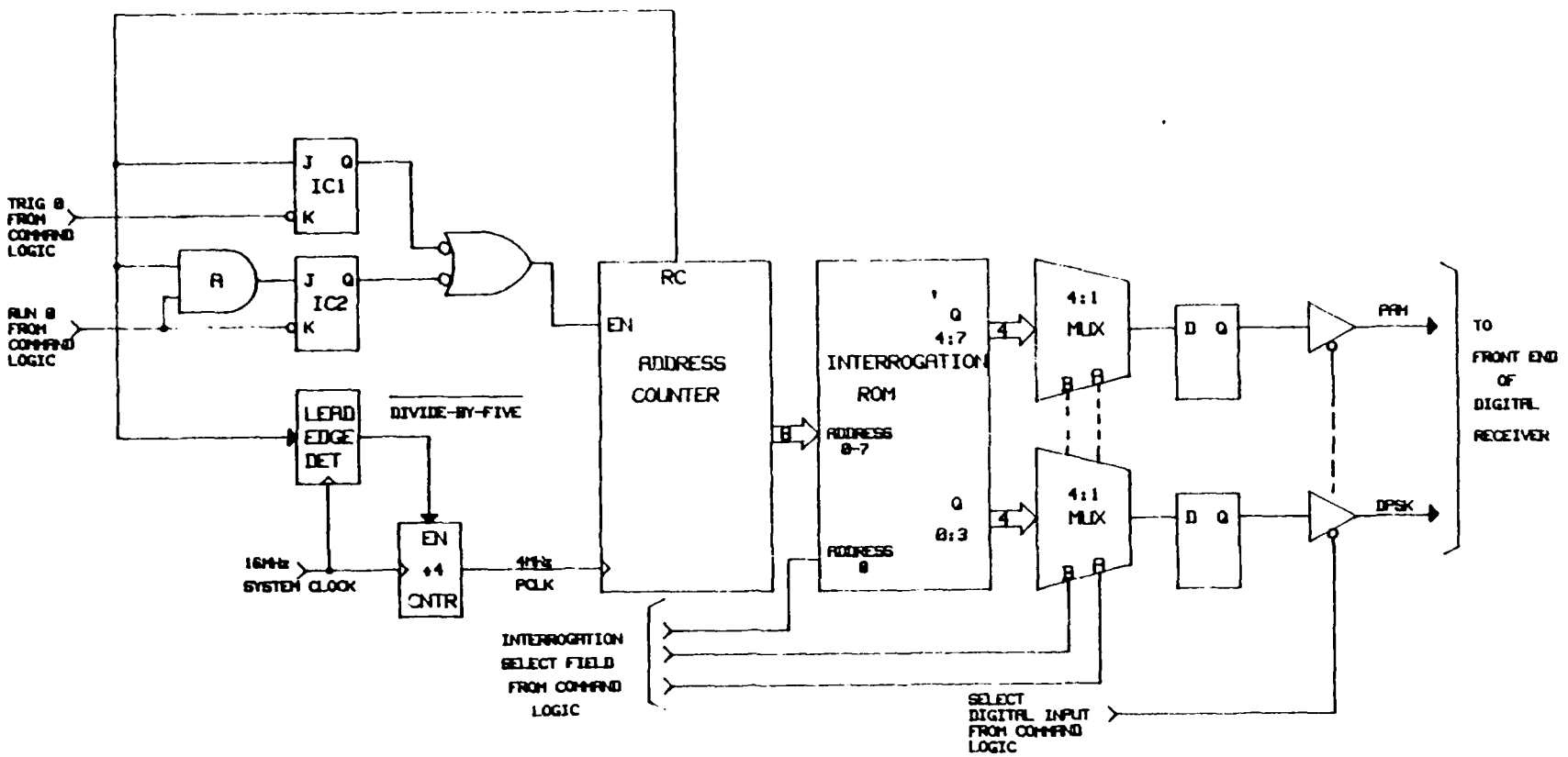


FIGURE 6.2.2.7-1. INTERROGATION SIMULATION CIRCUITRY

When the continuous run signal (RUNO) goes active low, J/K flip-flop IC_2 is set enabling the ROM address counter. Similar to the single shot mode, the counter starts at 0 and increments to its maximum count. At which time, the ripple carry pulse is generated. On the following clock lead edge, the counter is set back to 0 but flip-flop IC_2 remains set since AND gate A is disabled by the continuous run signal. Therefore, the counter remains enabled and the sequence is repeated at the rate of one interrogation approximately every 64 μs.

The simulated interrogation generation rate is actually one every 64.0625 μs. This is accomplished by lead edge detecting the ripple carry pulse of the ROM address counter and feeding it back to the divide by four counter forcing it to divide by five at that time. The purpose of this action is to assist in the verification of the complete TOAR counter logic in a fast efficient manner. By generating simulated interrogations at this rate, the TOAR counter can be sampled and checked for an incremental increase of 401_{HEX} after each received Roll-Call interrogation.

6.3 AZIMUTH DECODER/SIMULATOR.

Antenna azimuth data is normally transmitted from the antenna to the Mode S sensor in the form of ACPs provided by a shaft encoder. The 16,384 ACPs occur for each revolution of the antenna and a single ARP indicates when the antenna rotates past a reference azimuth, such as true north. ARIES is capable of testing a Mode S sensor whether it is operating with an antenna or not. When the sensor is operating without an antenna, ARIES generates simulated ACPs and ARPs just as if they were coming from the antenna. The rotation rate can be adjusted to within 250 ns of any desired revolution between 3.5 to 15 seconds. When the Mode S is coupled to an antenna, the antenna generates ACPs and ARPs. These signals are brought to the ACP Decoder Unit, so that it too will be aware of the true antenna azimuth and will be able to relate simulated replies and fruit to the actual antenna azimuth.

In both azimuth modes, the ACP Decoder generates 14-bit parallel azimuth words in synchronization with the Mode S ACPs and ARPs so that each interrogation received may be assigned a specific azimuth. These words are sent from the ACP Decoder to the Azimuth Register in the Uplink Receiver. The STU receives corrected ACPs and ARPs signals from the Azimuth Decoder/Simulator for its own use. A block diagram of the Azimuth Decoder/Simulator is illustrated in figure 6.3-1. This logic is mounted on the Uplink Receiver board located in slot No. 11 of the ARIES digital chassis.

6.3.1 Sensor Mode.

The azimuth source, either antenna or simulated, is selected via an Uplink Receiver command. When external azimuth is selected, ARIES azimuth words are slaved to the ACPs received from the Mode S antenna. In this case, multiplexer M_1 passes the ACPs and ARPs received from the antenna to the Mode S sensor. The serial ACPs are also passed to the input of a 14-bit serial-to-parallel converter via multiplexer M_2. The output of multiplexer M_2 is determined by the Uplink Receiver channel selection logic covered in section 6.2.2.1. The converter's output is true north corrected via addition of a manually programmable constant and transmitted to the Uplink Receiver Azimuth Register.

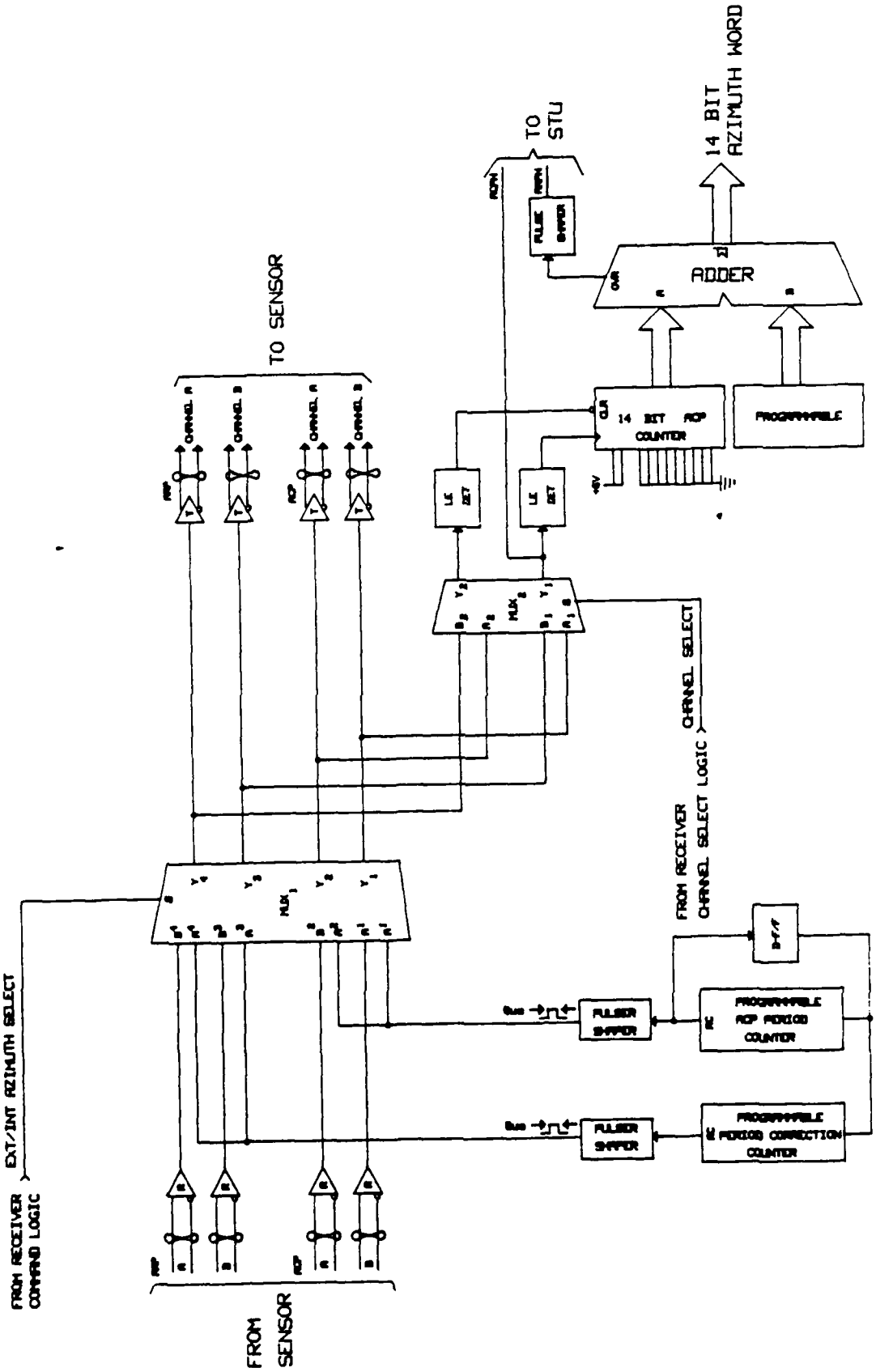


FIGURE 6.3-1. AZIMUTH DECODER/SIMULATOR BLOCK DIAGRAM

6.3.2 ARIES Mode.

When internal azimuth is selected, ARIES originates serial ACPs and ARPs. In this case, multiplexer M_1 passes the simulated ACPs and ARPs to the Mode S sensor. The simulated ACPs and ARPs are also passed to the input of a 14-bit serial-to-parallel converter, true north corrected and fed to the Uplink Receiver Azimuth Register.

6.3.2.1 Internal Azimuth Rate Programming.

The simulated azimuth rate generated by the ARIES is programmed by switch settings on the ACP Period Counter and the Period Correction Counter. These two counters operate in unison to produce scan rates from 3.5 seconds to 15 seconds in increments of 250 ns. The ACP Period Counter clocked at the 4-MHz clock rate produces an ACP period granularity of 250 ns. Therefore, this counter alone cannot simulate scan rates in which the ACP period is not a multiple of 250 ns. To simulate these scan rates, the ACP Period Counter is forced to skip one clock per ACP period increasing the period by 250 ns for an appropriate number of ACPs. In this way an average period finer than 250 ns is achieved over a complete revolution. The Period Correction Counter controls the number of ACPs per revolution generated with the one clock extended period.

Figure 6.3.2.1-1 shows a block diagram of the ACP simulation logic located on the Uplink Receiver board. The ACP Period Counter generates the ACP triggering pulses at a period controlled by the counters program switches. When an ARP is received, the Period Correction Counter is preset to the number set on its program switches. Its output Q^{14} goes low placing a high on OR gate A and setting the D flip-flop IC_1 to continuously enable the ACP Period Counter. Each time an ACP is generated, the Period Correction Counter is incremented by 1. When the Period Correction Counter reaches its maximum value, Q^{14} goes high preventing any further action by the counter and placing a low on gate A. Now the ACP Period Counter is controlled by the second input of OR gate A.

When the ACP Period Counter reaches its maximum value, the ACP trigger goes high placing a low on OR gate A disabling it. On the following clock cycle, the ACP Period Counter is preset, as usual, but this time the D flip-flop goes low inhibiting the counter for one clock cycle.

The following procedure is used to calculate the settings for the ACP Period Counter and the Period Correction Counter to program the Azimuth Simulator. An example is given along with the procedure to illustrate the calculations required for a 4.0-second scan rate.

a. Calculate the average period between ACP's using the following equation:

X (sec/ACP) = Scan Rate (sec/rev) / 16,384 (ACP's/rev)

Example: X = 4.0 sec/16,384 ACP's = 244.140625 μs/ACP

b. Truncate the average period to the nearest multiple of 250 ns (One 4-MHz clock cycle).

Example: $X_{truncated}$ = 244.000 μs

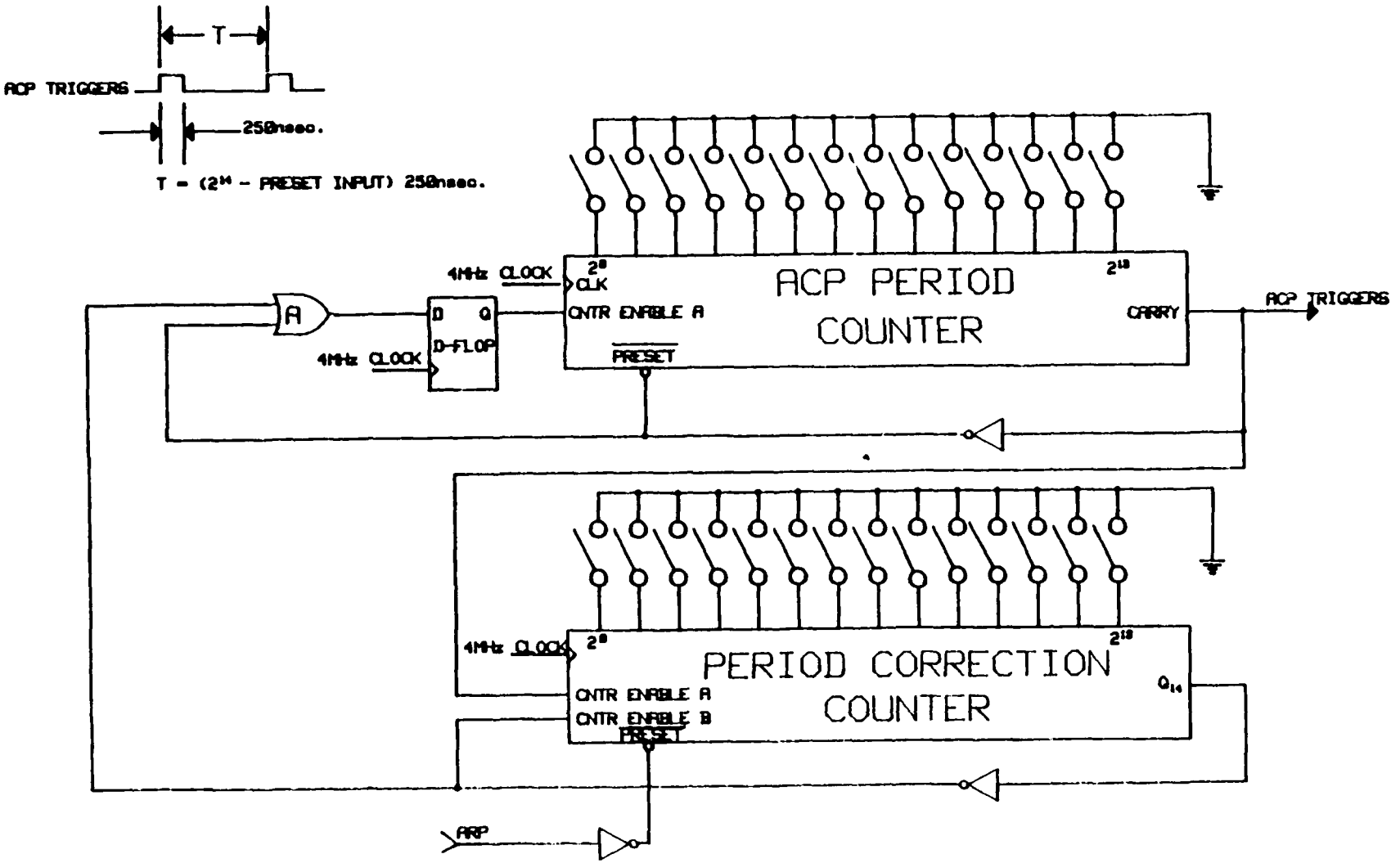


FIGURE 6.3.2.1-1. ACP SIMULATION LOGIC BLOCK DIAGRAM

c. Calculate the number of clock cycles equivalent to the period shown in Step "b."

Example: $X_{clocks} = 244.000\ (\mu s)/0.250\ (\mu s/clock) = 976$ clocks

d. Convert the number of clock cycles calculated in Step "c" to a hexidecimal number. The Two's Complement of this number is set in the ACP Period Counter program switches location at 1F and 3F on the Uplink Receiver board. Switch No. 1 at location 1F is the MSB. (To program a 1, set the switch to its open position; to program a 0, set the switch to its close position.)

Example: $X_{clocks(Hex)} = 3D0$

Two's Complement of $X_{clocks(Hex)} = 3C30$

e. Calculate the number of extended ACP periods required for the desired scan rate using the following equation:

$$X_{truncated}(16,384 - Y) + (X_{truncated} + 0.25\ \mu s)Y = \text{Scan Rate}$$

Example: $244.0\ \mu s(16,384 - Y) + 244.25\ \mu s(Y) = 4.0$ sec

$$0.25\ \mu s(Y) = 4.0 \text{ sec} - (16,384 \times 244.0\ \mu s)$$

$$Y = 2304 \times 10^{-6}/0.25 \times 10^{-6} = 9216$$

f. Convert the number calculated in Step "e" to a hexidecimal number. This number is set on the Period Correction Counter program switches located at locations 5F and 7F with switch No. 1 in location 5F being the MSB.

Example: $Y_{(Dec)} = 9216 \Rightarrow Y_{(Hex)} = 2400$

6.3.2.2 North Mark Adjustment Procedure.

The Azimuth Decoder/Simulator produces a 14-bit parallel azimuth word from ACP and ARP signals received for use by the rest of the ARIES system. The azimuth word is adjustable over the full azimuth range in steps of 1 azimuth unit (AU) for the purpose of aligning the azimuth information to true north.

The North Adjustment circuit shown in figure 6.3-1, consists of a 2-input 14-bit adder with a 14-bit switch array feeding one input. A 14-bit ACP counter feeds the second input of the adder. The switch array provides the ACP offset to adjust the azimuth word.

The ACP counter is preset to 0 each time an ARP is lead-edge detected. The counter is then incremented by one each time an ACP is lead-edge detected. The ACP count, plus the offset, on the program switches are added to produce the 14-bit azimuth word.

The following procedure is used to calculate the ACP offset setting for the North Mark Adjustment circuit. An example is given along with the procedure to illustrate the calculations for an azimuth alignment from magnetic north to true north.

a. Determine the separation in degrees between magnetic north and true north and convert the number to ACPs.

Example: Assume magnetic north is 11^{o} greater than true north referenced in a clockwise direction.

Azimuth Offset = 11 (degrees) x 16,384/360 (ACPs/degrees)

= 500.622··· ACPs

b. Round off the Azimuth Offset to the nearest ACP and convert the number to a hexidecimal number. This number is set in the North Adjustment program switches located at location 7F and 9F with switch No. 1 of 9F being the MSB.

Example: $\text{Azimuth Offset}_{\text{rounded}} = 501$ ACPs

$\text{Azimuth Offset}_{(\text{Hex})} = 1F5$

6.3.3 Azimuth Error Detection.

The Azimuth Validation logic verifies that only one ARP exists for every 16,384 ACPs detected. This is accomplished by the logic shown in figure 6.3.3-1. The logic checks for two possible error conditions. IC_1 and gates A, B, and C are used to detect the condition that an ARP was missing after 16,384 ACPs occurred since the last detected ARP. IC_2 and gates D and E are used to detect the condition that an ARP was generated before 16,384 ACPs were received since the last detected ARP.

The ACP and the ARP signals, along with the ripple carry (CAR1) signal of the ACP counter, are the three inputs used by this logic. The ACP and the ARP signals are fed through lead-edge detectors to synchronize them to the 4-MHz system clock. The ACP lead edge pulse ACP-LE1 increments the ACP counter by one. The ACP counter generates the CAR1 pulse on the 16,384th ACP since the last detected ARP. It is at this moment, before the next ACP is detected, that an ARP is anticipated.

Assuming that 16,384 ACPs were counted since the last ARP was detected, the CAR1 signal goes active high setting up the window for the anticipated occurrence of the ARP. AND gate A and NAND gate C are enabled. When the ARP is detected, the trail-edge pulse ARP-TE1 is generated and fed to J/K flip-flop IC_1 via gate A. On the following clock cycle, IC_1 is set, preventing the Missing ARP error from being set on the 16,385th received ACP, by disabling gate C. Also, NAND gate B is enabled. When the lead-edge pulse ACP-LE1 is generated, it is fed to AND gate F via gate B generating the azimuth strobe pulse ARP-ST0. On the following clock cycle, IC_1 is reset and the ACP counter is set back to 0. This is the correct mode of operation.

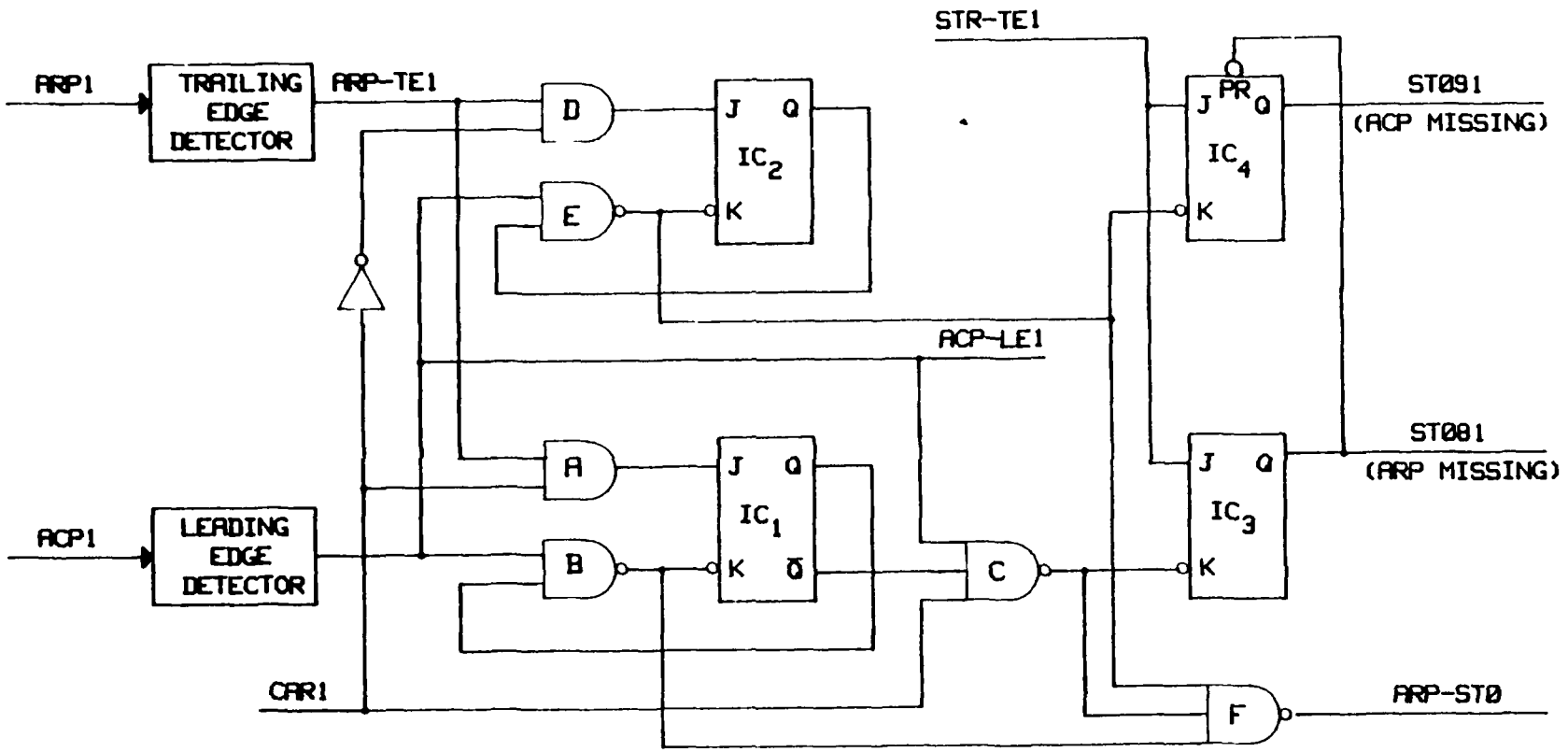


FIGURE 6.3.3-1. AZIMUTH ERROR DETECTION LOGIC DIAGRAM

If the ARP is not detected when the CAR1 signal is active high, IC_1 remains reset keeping NAND gate C enabled when the 16,385th ACP is received. At this time, the ACP-LE1 pulse is fed to J/K flip-flop IC_3 via gate C and to the ACP counter via AND gate F. On the following clock cycle, IC_3 is set, flagging that the ARP was missed, and the ACP counter is set back to 0.

If the ARP is detected when the CAR1 signal is inactive low, the ARP-TE1 pulse is fed to J/K flip-flop IC_2 via AND gate D. On the following clock cycle, IC_2 is set, enabling NAND gate E. When the next ACP is received, the ACP-LE1 pulse is fed to J/K flip-flops IC_2 and IC_4 via gate E and to the ACP counter via gate F. On the following clock cycle, IC_2 is reset, back to its initial state, IC_4 is set, flagging that the ARP was received early, and the ACP counter is set back to "0."

J/K flip-flops IC_2 and IC_4 automatically reset when the Uplink Receiver device (error) status is read by the processor.

6.4 REPLY GENERATORS.

The ARIES contains two reply generators; one handling the modeled replies based on a target scenario (the MRG), and the other handling the fruit replies based on a fruit environment scenario (the FRG). Each generator is under control of the computer via its own interface, and each uses its own microprocessor for routing and constructing replies. Each generator is capable of generating ATCRBS and Mode S replies.

The MRG receives reply data from the computer and the FRG from its two random process generators (ATCRBS and Mode S). Each reply generator directs, via microprocessor control, its set of target reply generators (the MRG contains a set of three target reply generators and the FRG contains a set of four target reply generators) to create properly formatted and sequenced modulation control signals. These signals modulate the IF Unit of each target reply generator with the carrier frequencies off set by 500 KHz centered around 60 MHz. The modulated output of all seven target reply generators are then combined in the IF Combiner and fed through an RF mixer to produce a 1090-MHz output. This output is injected into the Mode S sensor ARIES input port.

In the descriptions that follow, the details of operation of the MRG digital circuitry (section 6.4.1) precede the details of operation of the FRG digital circuitry (section 6.4.2). However, each reply generator includes a controller and each includes a set of target reply generators, the operations of which are very similar. In many cases they are identical except for the "modeled reply" and "fruit reply" modifying terminology and stored control programs. The hardware components of the MRG controller and the FRG controller are identical. Also the hardware comprising the target reply generators are identical. Description of the identical modules are given once under section 6.4.1, the Modeled Reply Generator, and differences only under section 6.4.2, the Fruit Reply Generator. Also, a similar controller is used by the RRG, the differences of which are described in section 6.8. The analog circuitry for both reply generators is given in section 6.4.3.

6.4.1 Modeled Reply Generator.

The MRG consists of the MRC, three target reply generators, referred to as the MAT generators, and the MRG Buffer Interface. A block diagram of the major modules of the MRG is presented in figure 6.4.1-1.

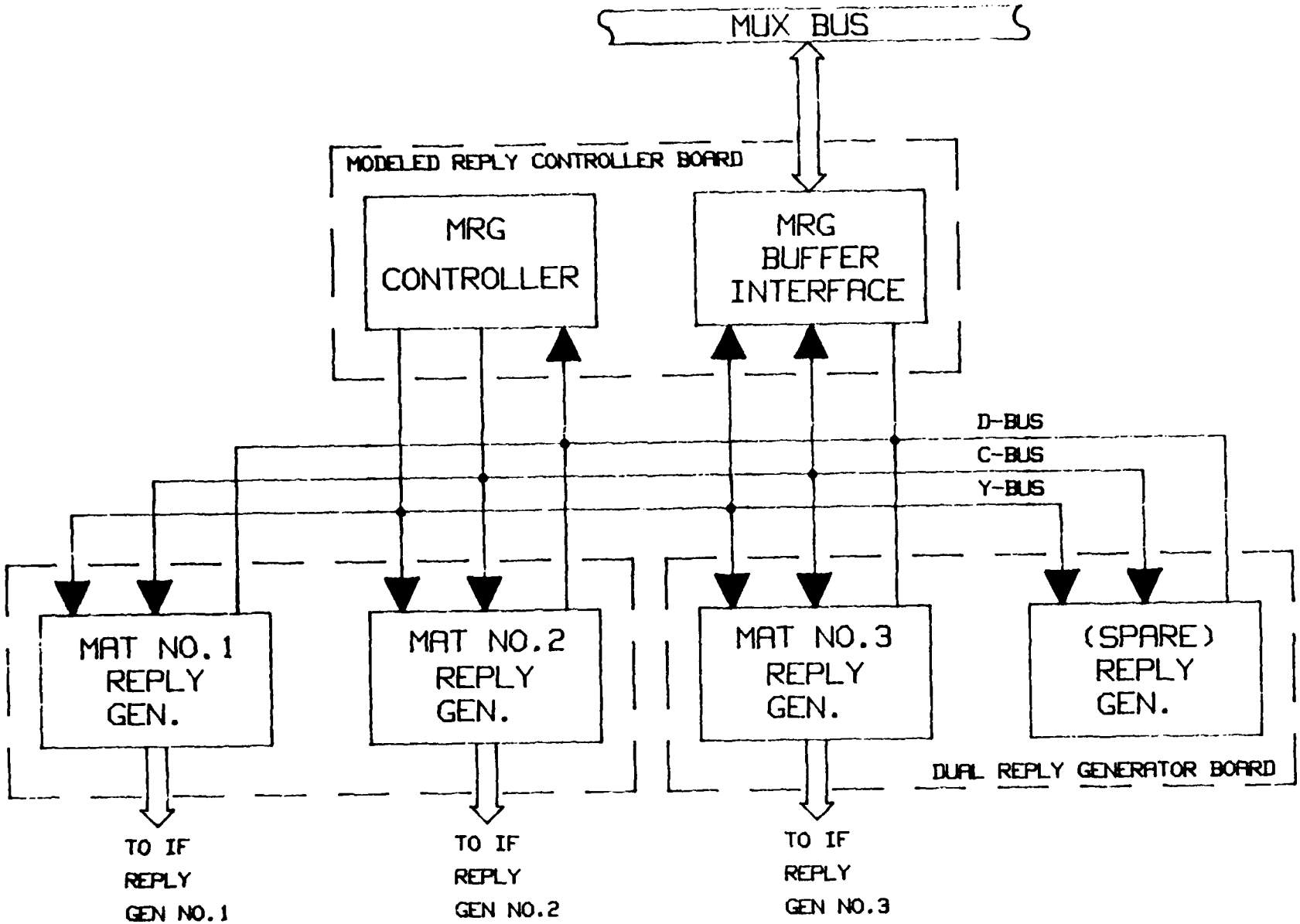


FIGURE 6.4.1-1. MRG DIGITAL CONFIGURATION

6.4.1.1 Modeled Reply Controller.

The primary function of the MRC, a high-speed bit-slice microprocessor, is to read reply data blocks from the MRG Buffer Interface and distribute these blocks between the three available reply generators. The controller maintains its control program (microcode) in ROM. This permits higher speed operation and the flexibility of changing the microinstruction set, by altering the ROM array, without modifying hardware. A detailed description of the microinstruction set and the microcode routines used is given in appendix B.

Understanding the operations of the MRC is best understood by examining its block diagram shown in figure 6.4.1.1-1. The controller can be separated into the following six functionally related sub-blocks:

a. Microprocessor
b Branch Control Logic
c. Program Controller
d. Microprogram Storage and Pipeline Register
e. Input Circuits
f. Output Circuits

The description on the operations of each block is given following the description of the controller overall architecture.

6.4.1.1.1 Controller Architecture.

The controller is basically a bit-slice microprocessor designed around four 2900 bipolar microprocessor components: (1) the 2901B 4-bit cascadable microprocessor slice, the 2902A high speed look-ahead carry generator, (2) the 2904 status and shift control unit, and (4) the 2910A program controller.

Four 2901Bs were used to form a 16-bit word microprocessor. These large scale integration (LSI) chips consist of a 16-word by 4-bit two port RAM, a high speed arithmetic logic unit (ALU), and associated shifting, decoding, and multiplexing circuitry. Next address selection and program control uses the 2910A to point to any one of 512 locations in the microcode (ROM). The 2904 provides the necessary shift linkage, multiplexers, status registers, condition code multiplexing, and other miscellaneous functions which are usually performed in medium scale integration (MSI) around an ALU. The input multiplexer/register allows multiplexing of input data from the microprogram memory or D-Bus. In addition, branch addresses stored in ROM can be input to the 2910A through the input multiplexer/register. The output circuitry allows the output data of the 2901B array to be placed on the Y-Bus or C-Bus. This circuitry consists of an output register for the data output to the Y-Bus, and a control register for command output to the C-Bus. All clocked circuitry uses the 4-MHz system clock.

6.4.1.1.2 The Microprocessor.

The heart of the MRC is the 2901B microprocessor slice. This 4-bit device, as shown in figure 6.4.1.1.2-1, consists of a 4-bit x 16-word two-port RAM, a high speed ALU, and associated shifting decoding and multiplexing circuitry. The 9-bit microinstruction word is organized into three groups of three bits each and selects the ALU source operands, the ALU function, and the ALU destination register.

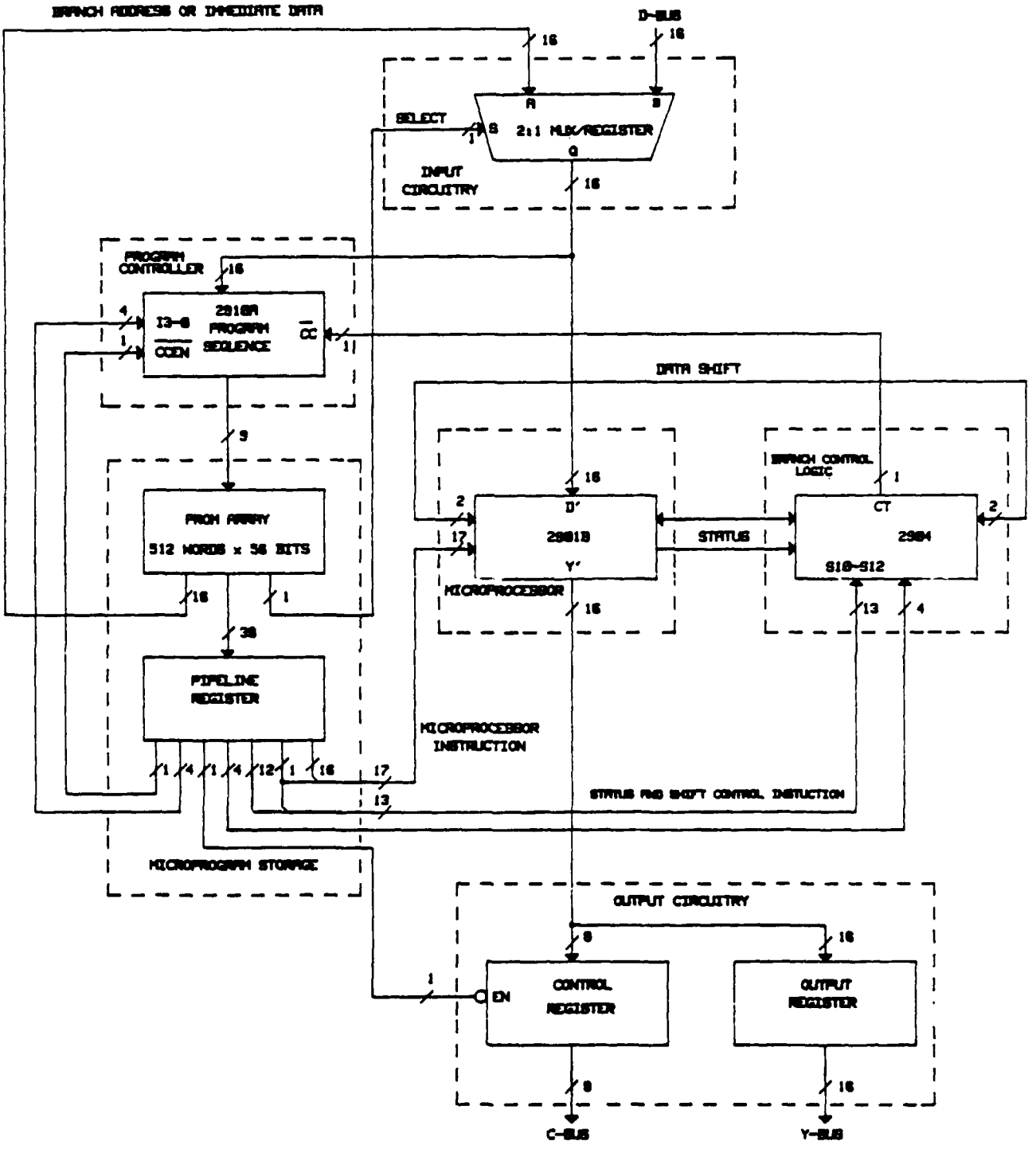


FIGURE 6.4.1.1-1. MRC BLOCK DIAGRAM

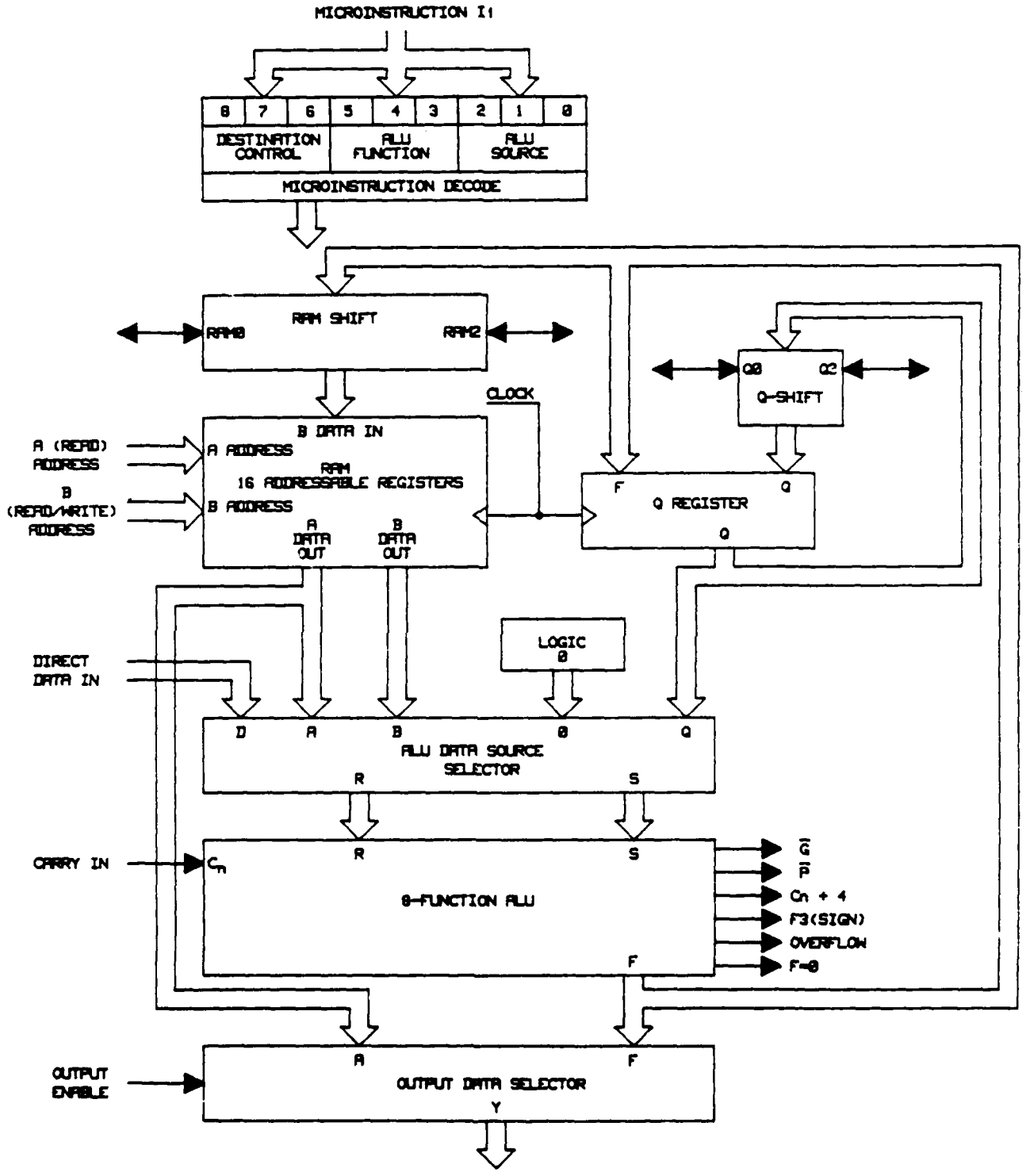


FIGURE 6.4.1.1.2-1. MICROPROCESSOR SLICE BLOCK DIAGRAM

Four cascaded 2901Bs implement the 16-bit microprocessor. Other support circuitry used are the 2902A look-ahead carry generator for the high-speed arithmetic operations, and the instruction decode shift multiplexer for the arithmetic and logic shift and rotate operations as shown in figure 6.4.1.1.2-2. The instruction decode shift multiplexer and the carry-in multiplexer are parts of the 2904 status and shift control unit (see section 6.4.1.1.3 Branch Control Logic, for more detail).

Data in any of the 16 words of the RAM can be read from the A port of the RAM as controlled by the 4-bit A address field (A_{3-0}). Likewise, data in any of the 16 words of the RAM can be read simultaneously from the B port of the RAM as controlled by the 4-bit B address field (B_{3-0}). If the same address is applied on both address fields, identical data will appear on both the A and B data outputs simultaneously.

New data is written into any of the 16 word locations defined by the B address field of the RAM. During the process of writing, the A address field is ignored. The B data input of the RAM is driven by a three-input multiplexer used to shift the output data F of the ALU if desired, up 1 bit position, down 1 bit position, or no shift in either direction.

There is one additional register called the Q-register. It has its own three-input multiplexer which performs a similar role as the RAM's three input multiplexer to shift the output data of the ALU up 1 bit position, down 1 bit position, or no shift before the data enters the Q-register. The Q-register's primary function is for multiplication and division but it can be used as an accumulator or another holding register.

The R and S inputs of the ALU are driven by a two-input multiplexer and a three-input multiplexer, respectively. Both of the multiplexers have an inhibit state effectively passing no data. This is equivalent to selecting 0 as the source operand. This multiplexer scheme gives the flexibility of selecting various pairs as source operands as defined by microinstruction bits I_0, I_1, and I_2 shown in table 6.4.1.1.2-1. The R input multiplexer has the RAM's A data output and direct data D connected to its two inputs. Likewise, the S input multiplexer has the RAM's A data output, the RAM's B data output, and the Q-register output connected to its three inputs.

TABLE 6.4.1.1.2-1. ALU SOURCE CONTROL

MNEMONIC	MICRO CODE				ALU SOURCE	
	I_2	I_1	I_0	Octal Code	R	S
AQ	L	L	L	0	A	Q
AB	L	L	H	1	A	B
ZQ	L	H	L	2	0	Q
ZB	L	H	H	3	0	B
ZA	H	L	L	4	0	A
DA	H	L	H	5	D	A
DQ	H	H	L	6	D	Q
DZ	H	H	H	7	D	0

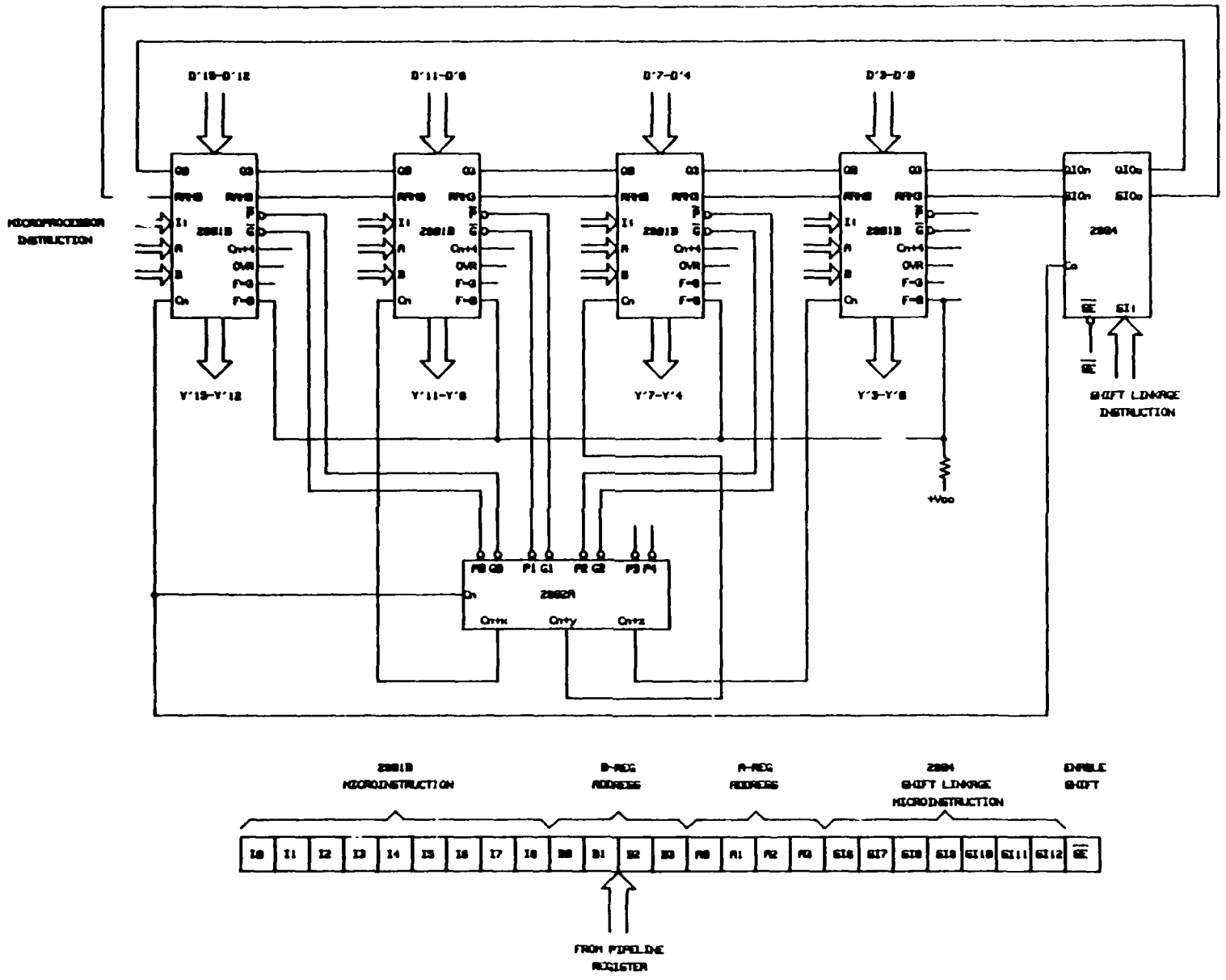


FIGURE 6.4.1.1.2-2. 16-BIT MICROPROCESSOR BLOCK DIAGRAM

The high-speed ALU can perform eight operations, three arithmetic, and five logic. These operations are performed on the two 4-bit input words R and S. The microinstruction bits I_3, I_4, and I_5 are used to select the ALU operation as shown in table 6.4.1.1.2-2. The octal code is shown for reference.

TABLE 6.4.1.1.2-2. ALU OPERATION CONTROL

MNEMONIC	MICRO CODE I_5	I_4	I_3	Octal Code	OPERATION
ADD	L	L	L	0	R Plus S
SUBR	L	L	H	1	S Minus R
SUBS	L	H	L	2	R Minus S
OR	L	H	H	3	R 'OR' S
AND	H	L	L	4	R 'AND' S
NOTRS	H	L	H	5	/R'AND'/S
EXOR	H	H	L	6	R'EX-OR'S
EXNOR	H	H	H	7	R'EX-NOR'S

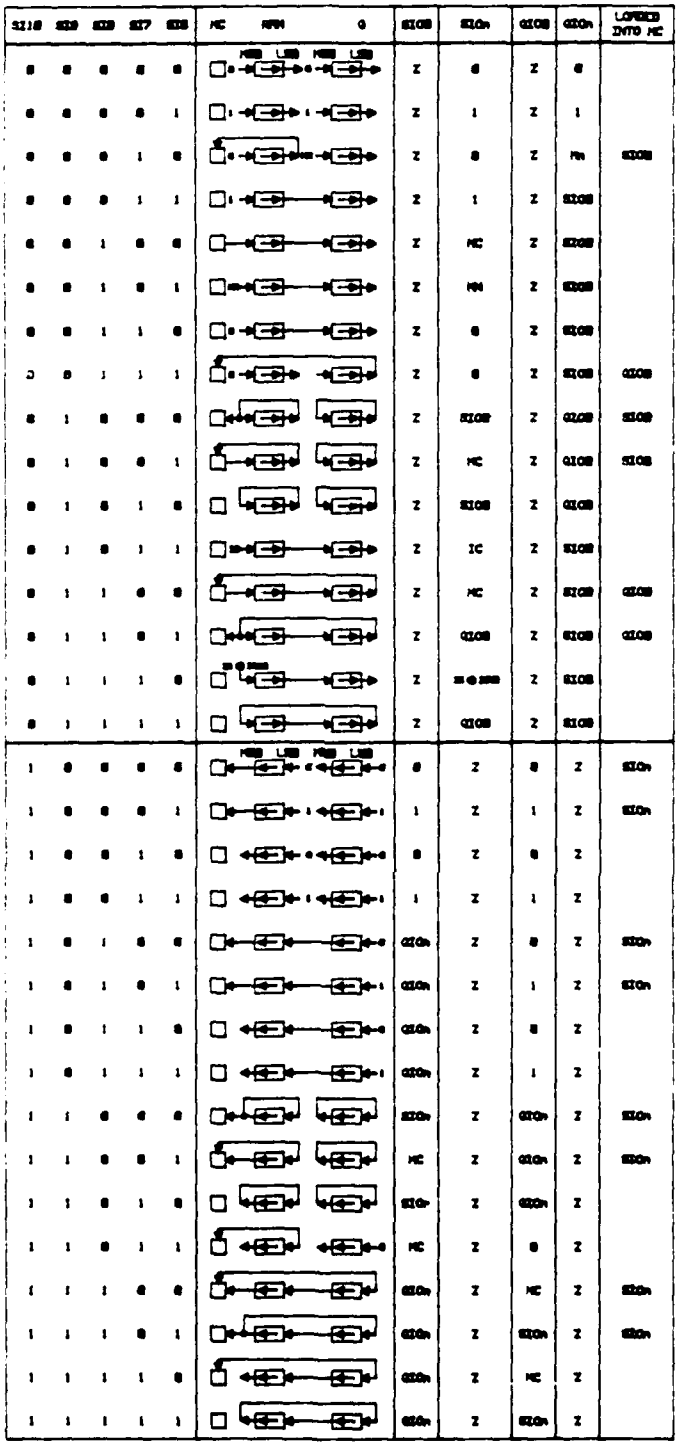
The direct input D allows external data to be entered into the microprocessor. The external data could be a load immediate constant stored in the microinstruction or data from the D-Bus (see section 6.4.1.1.5 Input Circuits, for more detail). These data can be loaded in any of the 17 registers (RAM and Q-register).

One additional input to the ALU is the carry-in bit, C_n. It is used in conjunction with the carry generate and the carry propagate outputs of the ALU to run the 2902A look-ahead carry generator. This arrangement allows the anticipation of carries across the 2901Bs, thereby eliminating the lengthy carry propagation time from the LSB to the MSB. The C_o output of the Carry-In Control Multiplexer that is part of the 2904 status and shift control unit is connected to the C_n of the least significant 2901B.

The output data F of the ALU can be routed as input to other modules via the Y-Bus, or as input to the RAM and Q-register. Also, the output can be shifted up 1 bit before it is loaded in the RAM or Q-register. Altogether, eight possible combinations exist, as controlled by the microinstruction bits I_6, I_7, and I_8 illustrated in table 6.4.1.1.2-3.

The ALU has four status-oriented outputs. They are: (1) the carry flag (C_{n+4}), (2) the sign bit (F_3), (3) the zero detect flag (F=0), and (4) the overflow flag (OVR). The F=0 is an open-collector output wired OR'ed between microprocessor slices and the branch control logic of the 2904 status and shift control unit. F=0 is high when all F outputs are low. The carry-out bit is set whenever an operation, either logical or arithmetic, exceeds the 16-bit range of the microprocessor.

In conjunction with the shift linkage multiplexer, the ALU can perform 32 different shift and rotate functions. Figure 6.4.1.1.2-3 shows the wide selection of input and output connections possible and the single and double length shift and rotate functions.



NOTES: 1. Z = High impedance (oututs off) state.

2. Outputs enabled and MC loaded only if SE is LOW.

3. Loading of MC from I10-6 overrides control from I3-0, $\overline{CEM}$, $\overline{EC}$.

FIGURE 6.4.1.1.2-3. SHIFT LINKAGE MULTIPLEXER INSTRUCTION CODE

TABLE 6.4.1.1.2-3. ALU DESTINATION CONTROL

MIRCO CODE				RAM FUNCTION		Q-REG. FUNCTION		Y	RAM SHIFTER		Q SHIFTER	
I_8	I_7	I_6	OCTAL CODE	SHIFT	LOAD	SHIFT	LOAD	OUTPUT	RAM_0	RAM_3	Q_0	Q_3
L	L	L	0	X	NONE	NONE	F-Q	F	X	X	X	X
L	L	H	1	X	NONE	X	NONE	F	X	X	X	X
L	H	L	2	NONE	F-B	X	NONE	A	X	X	X	X
L	H	H	3	NONE	F-B	X	NONE	F	X	X	X	X
H	L	L	4	DOWN	F/2-B	DOWN	Q/2-Q	F	F0	IN_3	Q_0	IN_3
H	L	H	5	DOWN	F/2-B	X	NONE	F	F_0	IN_3	Q_0	X
H	H	L	6	UP	2F-B	UP	2Q-Q	F	IN_0	F_3	IN_0	Q_3
H	H	H	7	UP	2F-B	X	NONE	F	IN_0	F_3	X	Q_3

X = Don't care.
B = Register address by B inputs.
UP is toward the MSB, DOWN is toward the LSB.

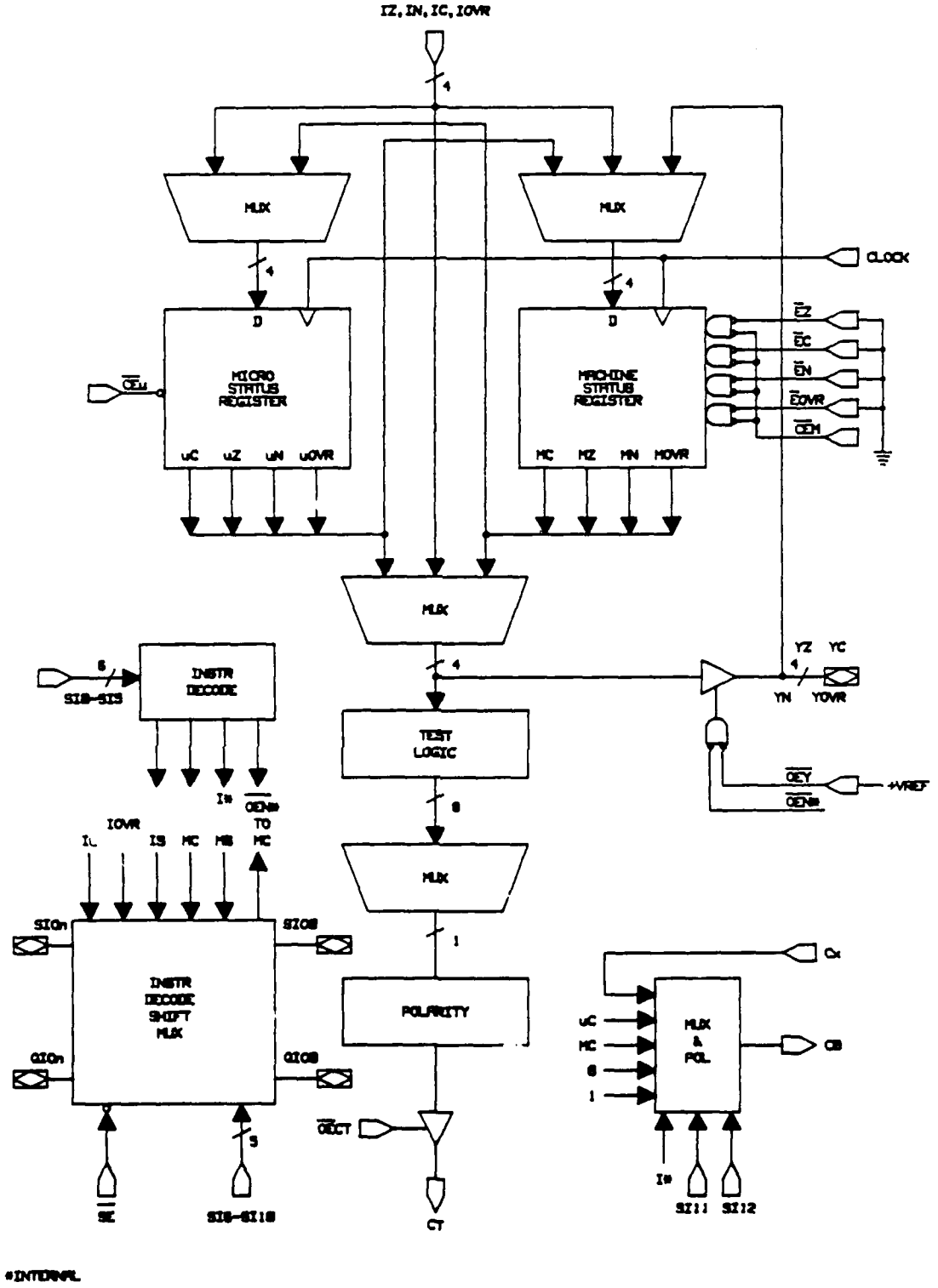
Referring to figure 6.4.1.1.2-2, the shift line RAM_3 of the least significant slice (RAM_3 being the MSB of the register stack) is tied to the shift line RAM_0 of the adjacent slice (RAM_0 being the LSB of the slice register stack). The remaining shift lines, RAM_3 of the most significant slice and RAM_0 of the least significant slice are tied to the shift input/output lines SIO_n and SIO_0 of the Shift Linkage Multiplexer, respectively. All of these lines are bidirectional tristate lines. The shift lines for the Q-register (Q_0 and Q_3) and shift input/output lines of the Shift Linkage Multiplexer (QIO_0 and QIO_n) are tied together in a similar manner.

The five shift control bits, SI_6 through SI_{10}, are used to control the Shift Linkage Multiplexer. The three control bits, I_6, I_7, and I_8, are used to control the shift functions of the slices. Microinstruction bit 23 is used by the slices as line I_7 and the Shift Linkage Multiplexer as line SI_{10}. This bit determines whether the shift is up or down.

The clock input to the 2901B controls the RAM, the Q-register, and the A and B data latches. When enabled, data is clocked into the Q-register on the low-to-high transition of the clock. When the clock input is high, the A and B latches are open and will pass whatever data is present at the RAM outputs. When the clock is low, the latches are closed and will retain the last data entered.

6.4.1.1.3 Branch Control Logic.

The Branch Control Logic is shown in figure 6.4.1.1.3-1 along with the Shift Control Multiplexer and the Carry-In Control Multiplexer which are nearly independent blocks of logic in the 2904 Status Shift and Control Unit. In this section, we will concentrate on the Branch Control Logic of the microcontroller.



*INTERNAL

FIGURE 6.4.1.1.3-1. STATUS AND SHIFT CONTROL UNIT BLOCK DIAGRAM

The Branch Control Logic contains two 4-bit registers used to store status outputs of the ALU:

a. Carry (I_c), the result of an arithmetic, logical, or shift operation in which a carryout is generated.

b. Negative (I_n), resulting from a negative (i.e., MSB = 1) arithmetic, logical, or shift operation.

c. Zero (I_z), if the result of an arithmetic, logical, or shift operation is zero.

d. Overflow (I_{ovr}), the result of an arithmetic, logical, or shift operation which exceeds the available two's complement number range.

There are two registers designated Machine Status Register (MSR) and Micro Status Register (μSR). Each register is independently controlled. Control lines $/E_z$, $/E_c$, $/E_n$, and $/E_{ovr}$ of the MSR are tied low, allowing MSR updating via the control line $/CE_M$. Updating of the μSR is controlled by $/CE_u$. Both control lines come from the pipeline register. When these lines are high, the status registers remain unchanged. The status registers are updated on the low-to-high transition of the 4-MHz system clock when their corresponding control lines are low.

The instruction code (SI_0 through SI_5) for the registers falls into two groups: (1) register operations, and (2) load operations. As mentioned previously, all operations require the appropriate control line to be active low to operate. Also, bit operations can be performed on the μSR but is not used. Table 6.4.1.1.3-1 shows all of the operations implemented in controlling the branch control functions.

6.4.1.1.4 The Microprogram Controller.

The microprogram controller used is the 2910A microprogram controller. This controller is an address sequencer capable of addressing up to 4096 words of microcode. (Currently, the size of the microcode memory is 512 words; however, it is designed for 1024 words without hardware modifications.) A block diagram of the program controller is shown in figure 6.4.1.1.4-1.

Referring to figure 6.4.1.1.4-1, the program controller contains a four input multiplexer for selecting one of four address sources. These sources are:

a. Direct Input
b. Microprogram Counter
c. Register/Counter
d. Stack File

The direct input D is used for branch addressing. The branch address is a 12-bit field of the microinstruction (refer to figure 6.4.1.1.5-1, bits 4 through 15) which is looped back to the program controller via the input circuitry during the execution of a BRANCH instruction. The program counter contains an address one greater than the previous address and is used for sequencial execution of microinstructions. The register/counter R is not used by any of the defined microinstructions and so is not used as an address source by any of the controllers. The stack file (SF) is used to provide a return address linkage when executing subroutines.

TABLE 6.4.1.1.3-1. BRANCH CONTROL FUNCTIONS

SI_{3-0} HEX	SI_3	SI_2	SI_1	SI_0	SI_5=SI_4=0	SI_5=0 SI_4=1	SI_5=1 SI_4=0	SI_5=SI_4=1
0	0	0	0	0	$(\mu_N \oplus \mu_{OVR}) + \mu_Z$	$(\mu_N \oplus \mu_{OVR}) + \mu_Z$	$(M_N \oplus M_{OVR}) + M_Z$	$(I_N \oplus I_{OVR}) + I_Z$
1	0	0	0	1	$(\mu_N \odot \mu_{OVR}) \cdot \bar{\mu}_Z$	$(\mu_N \odot \mu_{OVR}) \cdot \bar{\mu}_Z$	$(M_N \odot M_{OVR}) \cdot \bar{M}_Z$	$(I_N \odot I_{OVR}) \cdot \bar{I}_Z$
2	0	0	1	0	$\mu_N \oplus \mu_{OVR}$	$\mu_N \oplus \mu_{OVR}$	$M_N \oplus M_{OVR}$	$I_N \oplus I_{OVR}$
3	0	0	1	1	$\mu_N \odot \mu_{OVR}$	$\mu_N \odot \mu_{OVR}$	$M_N \odot M_{OVR}$	$I_N \odot I_{OVR}$
4	0	1	0	0	μ_Z	μ_Z	M_Z	I_Z
5	0	1	0	1	$\bar{\mu}_Z$	$\bar{\mu}_Z$	$\bar{M}_Z$	$\bar{I}_Z$
6	0	1	1	0	μ_{OVR}	μ_{OVR}	M_{OVR}	I_{OVR}
7	0	1	1	1	$\bar{\mu}_{OVR}$	$\bar{\mu}_{OVR}$	$\bar{M}_{OVR}$	$\bar{I}_{OVR}$
8	1	0	0	0	$\mu_C + \mu_Z$	$\mu_C + \mu_Z$	$M_C + M_Z$	$I_C + I_Z$
9	1	0	0	1	$\bar{\mu}_C \cdot \bar{\mu}_Z$	$\bar{\mu}_C \cdot \bar{\mu}_Z$	$\bar{M}_C \cdot \bar{M}_Z$	$\bar{I}_C \cdot \bar{I}_Z$
A	1	0	1	0	μ_C	μ_C	M_C	I_C
B	1	0	1	1	$\bar{\mu}_C$	$\bar{\mu}_C$	$\bar{M}_C$	$\bar{I}_C$
C	1	1	0	0	$\bar{\mu}_C + \mu_Z$	$\bar{\mu}_C + \mu_Z$	$\bar{M}_C + M_Z$	$\bar{I}_C + I_Z$
D	1	1	0	1	$\mu_C \cdot \bar{\mu}_Z$	$\mu_C \cdot \bar{\mu}_Z$	$M_C \cdot \bar{M}_Z$	$I_C \cdot \bar{I}_Z$
E	1	1	1	0	$I_N \oplus M_N$	μ_N	M_N	I_N
F	1	1	1	1	$I_N \odot M_N$	$\bar{\mu}_N$	$\bar{M}_N$	$\bar{I}_N$

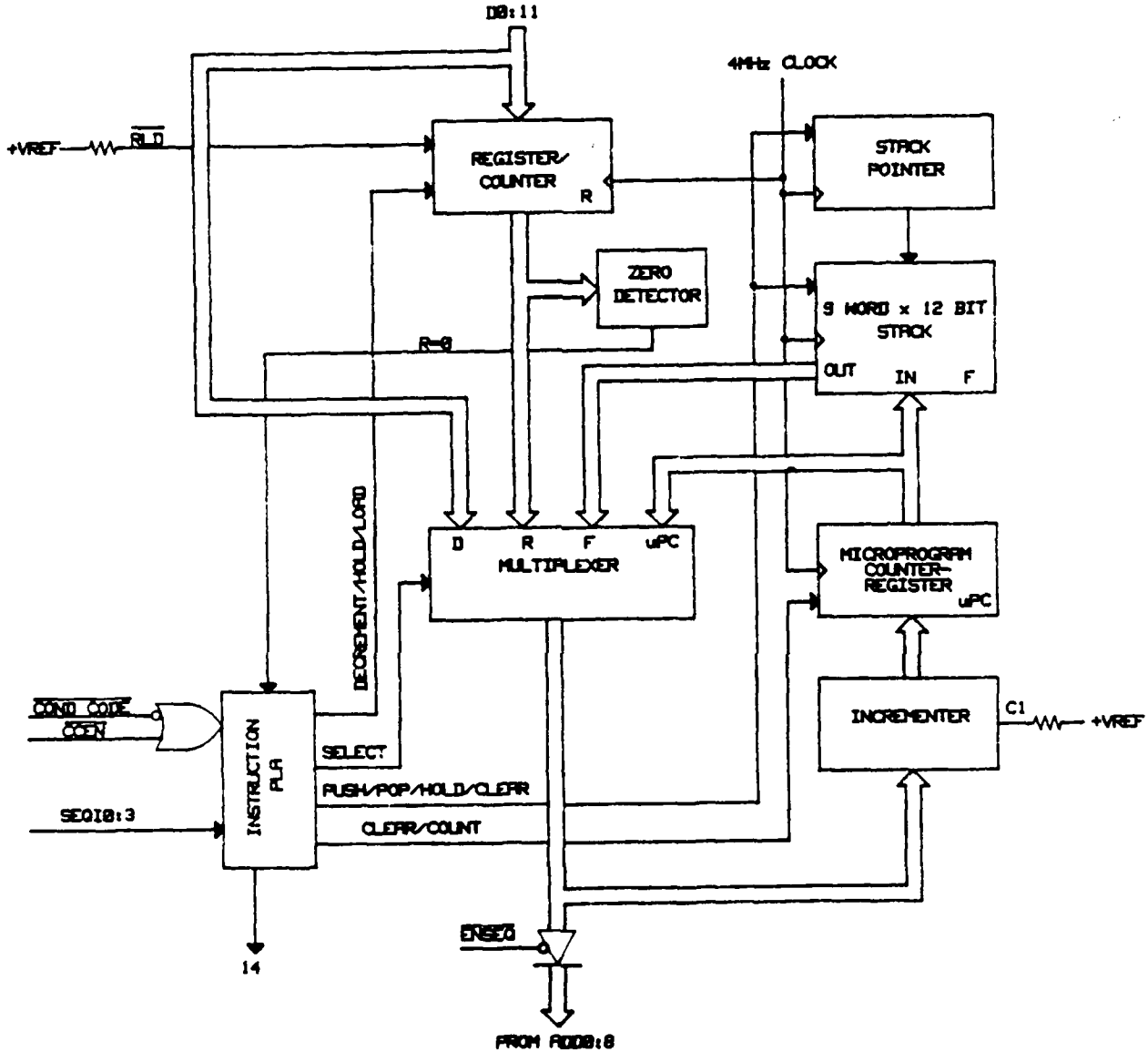


FIGURE 6.4.1.1.4-1. PROGRAM CONTROLLER BLOCK DIAGRAM

The source selected is determined by the microprogram controller instruction (I_i) and the code-condition (/CC) and code-condition enable (/CCEN) control lines. Table 6.4.1.1.4-1 shows the implemented instruction set of the 2910A program controller.

TABLE 6.4.1.1.4-1. MICROCONTROLLER INSTRUCTION SET

I_3-I_0	MNEMONIC	NAME	Y	STACK	Y	STACK	ENABLE
0	JZ	JUMP ZERO	0	CLEAR	0	CLEAR	PL
1	CJS	COND JSB PL	PC	HOLD	D	PUSH	PL
3	CJP	COND JUMP PL	PC	HOLD	D	HOLD	PL
10	CRTN	COND RETURN	PC	HOLD	SF	POP	PL
14	CONT	CONTINUE	PC	HOLD	PC	HOLD	PL

6.4.1.1.5 Program Storage and Pipeline Register.

The program for the controller is stored in seven 27S31 ROMs. These ROMs are 8-bits x 512-words with the worst case access time of 55 ns. If additional microcode depth is required, the 27S181's can be directly substituted for the 27S31's. These ROMs are 8-bits x 1024-words with the worst case access time of 60 ns and have the same pin layout and packaging as the 27S31. The exception of this is the upper address pin which is active instead of a no connection as is the case with the 27S31. This upper bit is prewired so only a plug in substitution is required.

The microinstruction length is 56 bits and contains all the necessary control signals for the 2901B's, 2904, 2910A, multiplexer/register, and control register. In the case of branch and load immediate instructions, the ROM array contains the microprogram branch address and constant data, respectively.

The remaining 39 bits of the ROM array is buffered by a pipeline register as shown in figure 6.4.1.1.5-1. The pipeline design places the ALU and the microprogram memory in parallel speed paths, permitting the microinstruction fetch to occur in parallel with the data operation. This approach greatly improves the overall throughput of the device. A typical cycle time waveform is depicted in figure 6.4.1.1.5-2.

6.4.1.1.6 Controller Input Circuits.

The input circuits (see figure 6.4.1.1.6-1) of the controller is basically a 16-bit wide 2:1 multiplexer/register. The output of the multiplexer/register is tied to the D'-inputs of the microprocessor 2901B's and the direct data inputs of the 2910A Microprogram Controller.

The D-Bus is tied to one of the inputs of the multiplexer/register to allow data from external modules to be received by the controller. Since the D-Bus is an open collector bus tied to many modules, only one module, addressed by the Controller via the C-Bus, is allowed to output data. (See section 6.4.1.1.7, Controller Output Circuits, for more information on the C-Bus.) The second input of the multiplexer/register is tied to the 16-bit field of the micromemory (ROM), designated as the constant data field.

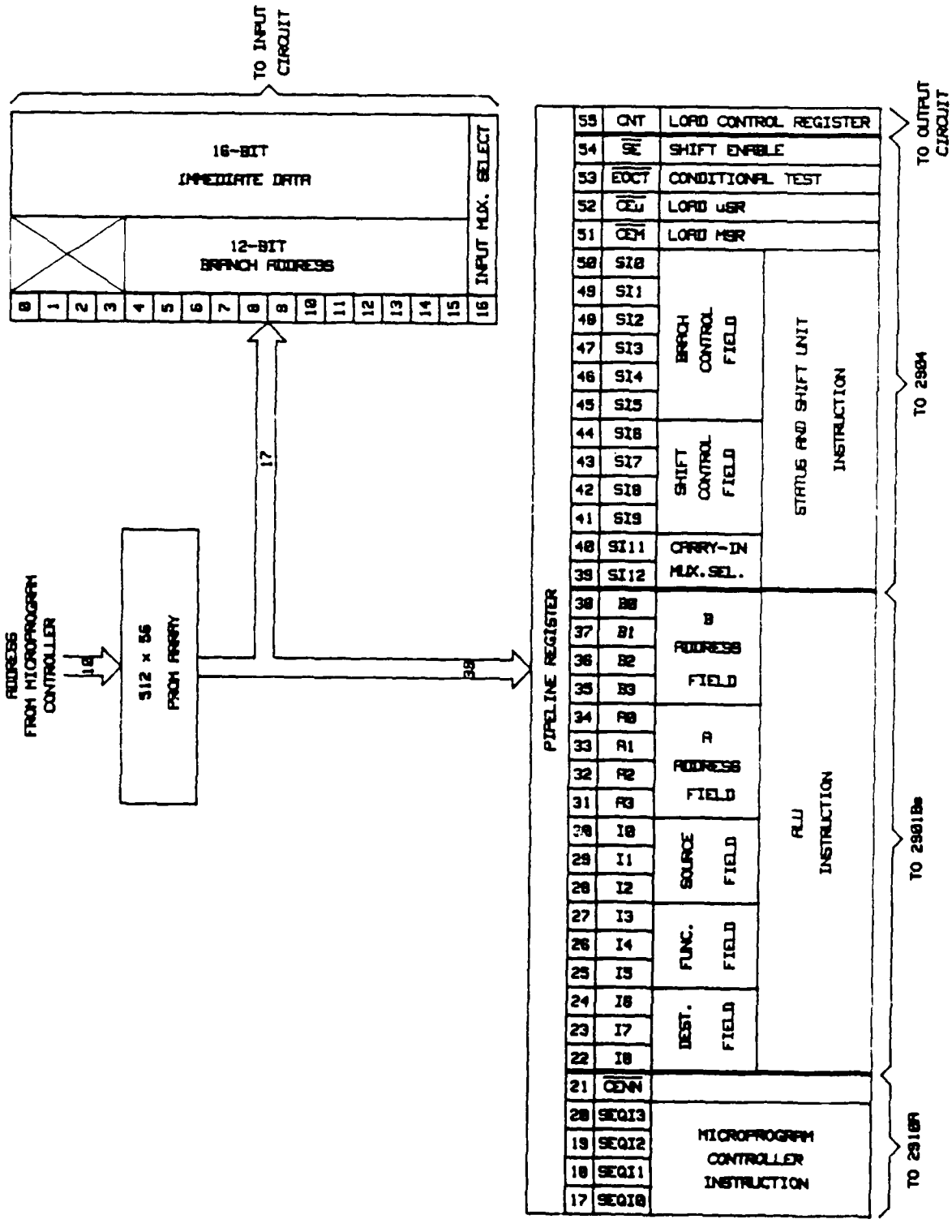


FIGURE 6.4.1.1.5-1. MICROPROGRAM STORAGE AND PIPELINE REGISTER

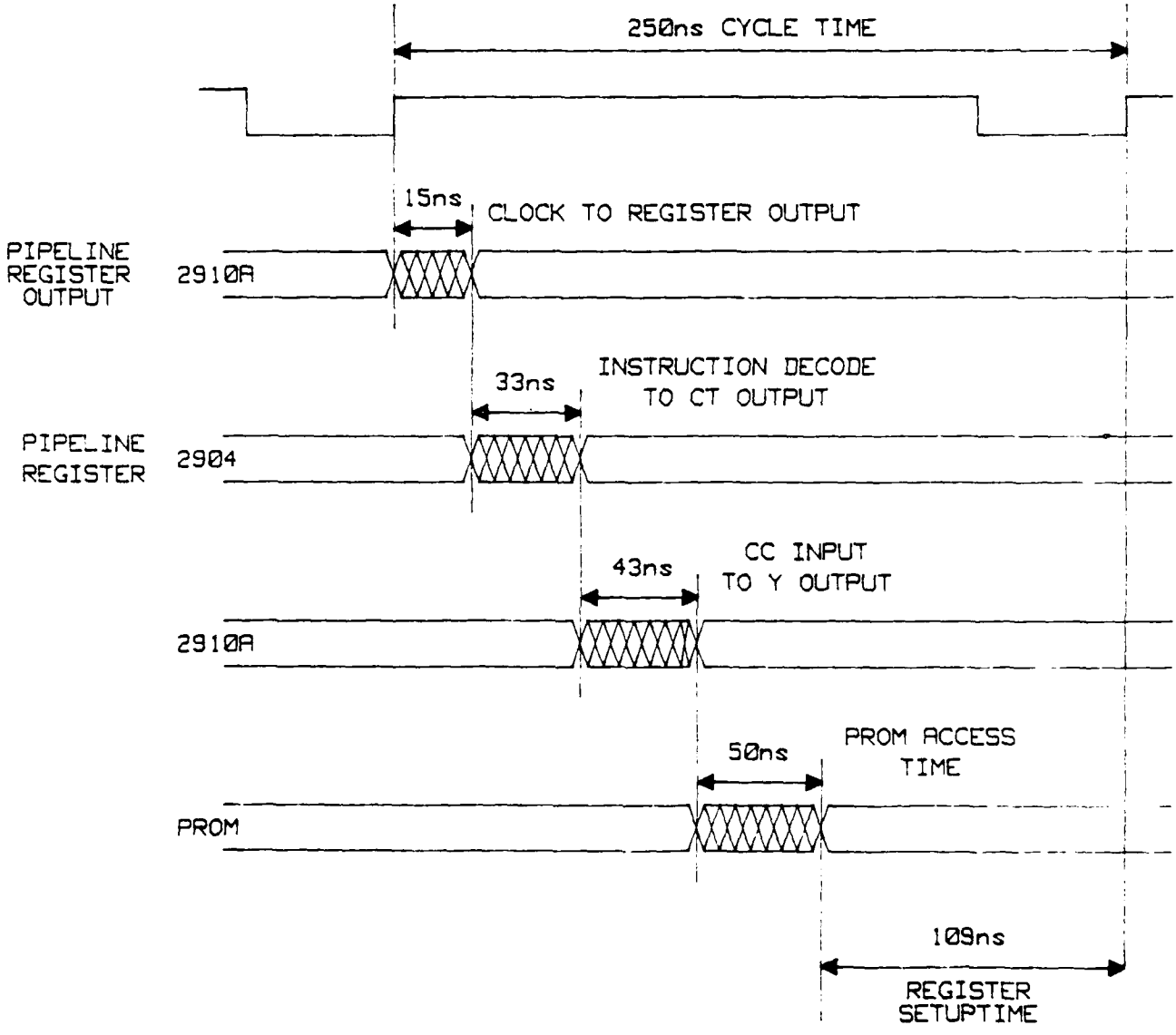


FIGURE 6.4.1.1.5-2. TYPICAL CYCLE TIMING WAVEFORM

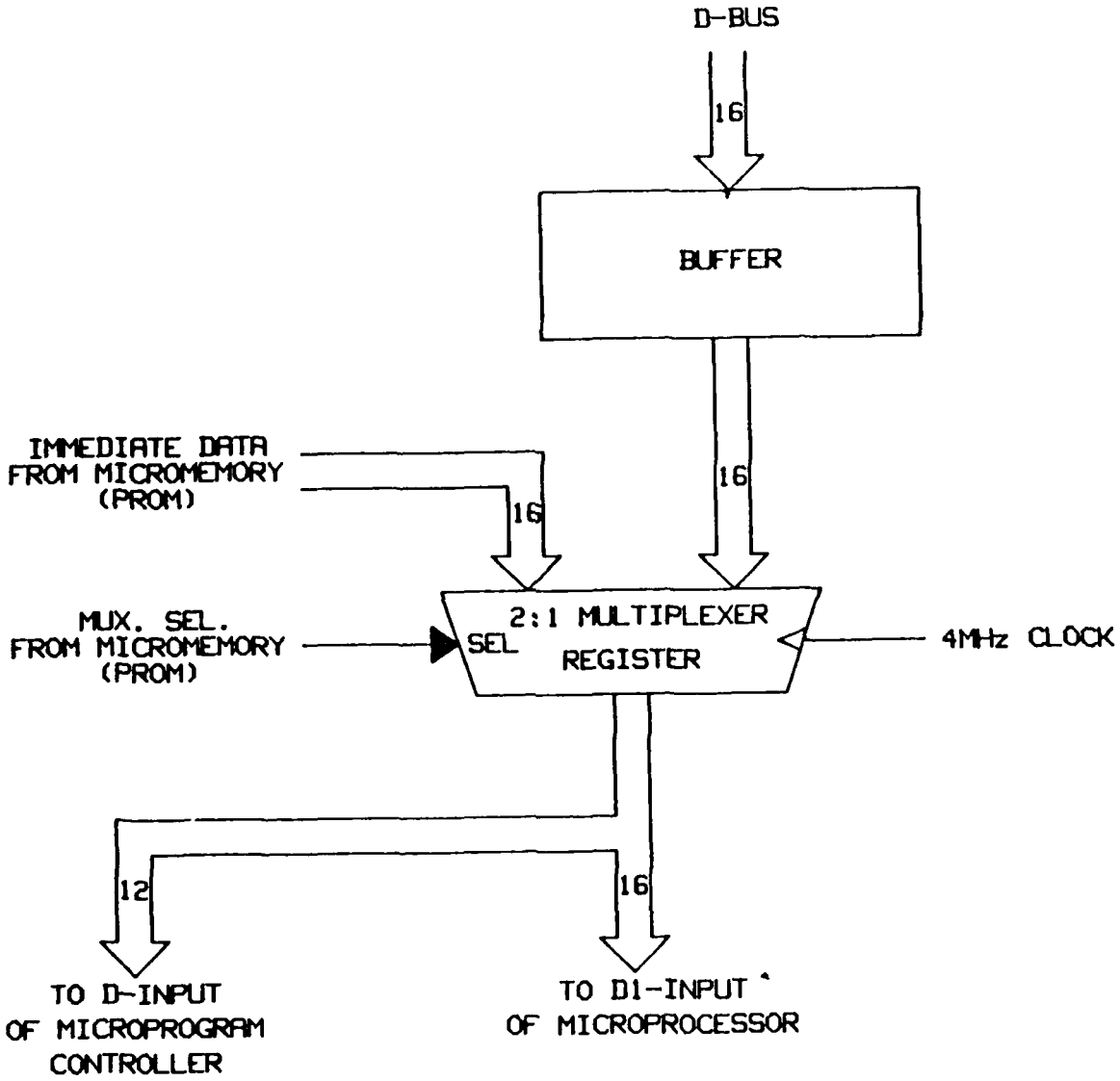


FIGURE 6.4.1.1.6-1. CONTROLLER INPUT CIRCUITS

Normally, the D-Bus is selected but during the execution of a branch or load immediate instruction, the constant data field is selected. The input selected is controlled by microinstruction bit 16.

6.4.1.1.7 Controller Output Circuits.

The outputs of the 2901B's Y'-Bus drive two sets of registers located in the output circuits as shown in figure 6.4.1.1.7-1. One of the registers is labeled as the output register, and the other is labeled as the control register. The 16-bit output register, which is always enabled, serves as a data link between the controller and the remaining modules of the generator, whether the generator is the MRG or the FRG.

The Y-Bus is tied to many modules. Therefore, in order to control each module properly, another register is needed. This is the purpose of the 8-bit control register. When enabled by a microinstruction control bit (microinstruction bit 55), the control register is loaded with a command consisting of two fields: (1) the module address (ADRS) code, and (2) operation (OP) code. The ADRS code a 4-bit field, directs only 1 module to respond to the OP code and the data on the Y-Bus. The command will remain in the control register until the next command is loaded. The operations and their codes for the MRG controller are shown in table 6.4.1.1.7-1.

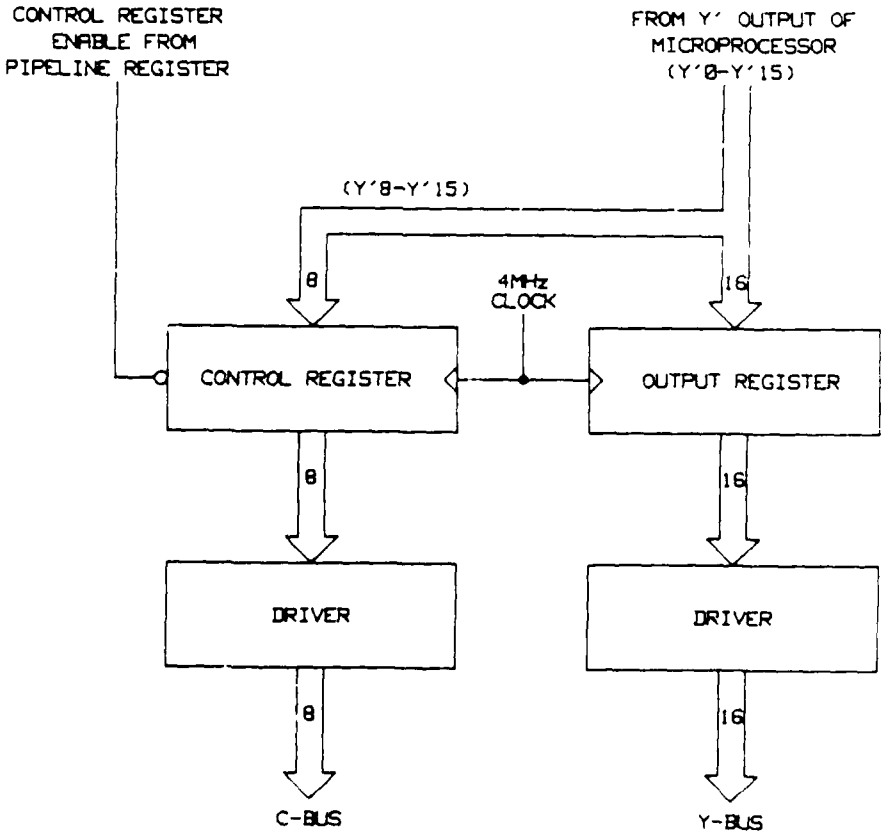


FIGURE 6.4.1.1.7-1. CONTROLLER OUTPUT CIRCUITS

TABLE 6.4.1.1.7-1. MRC MICROCODE COMMANDS

MODULE	OPERATION	MIRCOCODE MNEMONIC	MODULE ADDRESS	OP CODE
MAT#1	READ MAT#1 STATUS	MAT1STAT	0	0
	RESET MAT#1	MAT1RST	0	1
	LOAD 4 WORDS INTO MAT#1	MAT1LD4	0	2
	LOAD 6 WORDS INTO MAT#1	MAT1LD6	0	3
	READ 4 WORDS FROM MAT#1	MAT1RD4	0	4
	MAT#1 CW ON	MAT1CW	0	5
	MAT#1 CW OFF	MAT1NCW	0	6
	MAT#1 DIAGNOSTIC MODE	MAT1DIAG	0	7
MAT#2	READ MAT#2 STATUS	MAT2STAT	1	0
	RESET MAT#2	MAT2RST	1	1
	LOAD 4 WORDS INTO MAT#2	MAT2LD4	1	2
	LOAD 6 WORDS INTO MAT#2	MAT2LD6	1	3
	READ 4 WORDS FROM MAT#2	MAT2RD4	1	4
	MAT#2 CW ON	MAT2CW	1	5
	MAT#2 CW OFF	MAT2NCW	1	6
	MAT#2 DIAGNOSTIC MODE	MAT2DIAG	1	7
MAT#3	READ MAT#3 STATUS	MAT3STAT	2	0
	RESET MAT#3	MAT3RST	2	1
	LOAD 4 WORDS INTO MAT#3	MAT3LD4	2	2
	LOAD 6 WORDS INTO MAT#3	MAT3LD6	2	3
	READ 4 WORDS FROM MAT#3	MAT3RD4	2	4
	MAT#3 CW ON	MAT3CW	2	5
	MAT#3 CW OFF	MAT3NCW	2	6
	MAT#3 DIAGNOSTIC MODE	MAT3DIAG	2	7
MAT#4	READ MAT#4 STATUS	MAT4STAT	3	0
	RESET MAT#4	MAT4RST	3	1
	LOAD 4 WORDS INTO MAT#4	MAT4LD4	3	2
	LOAD 6 WORDS INTO MAT#4	MAT4LD6	3	3
	READ 4 WORDS FROM MAT#4	MAT4RD4	3	4
	MAT#4 CW ON	MAT4CW	3	5
	MAT#4 CW OFF	MAT4NCW	3	6
	MAT#4 DIAGNOSTIC MODE	MAT4DIAG	3	7
MRG BUFFER INTERFACE	READ MRG I/F STATUS	IFSTATUS	6	1
	READ 4 WORDS FROM MRG I/F	READ4	6	2
	READ 6 WORDS FROM MRG I/F	READ6	6	3
	LOAD 4 WORDS INTO MRG I/F	LOAD4	6	4
	DECREMENT REPLY COUNTER	DECRC	6	5
	ISSUE MRG INTERRUPT	MRGATN	6	7

6.4.1.2 Modeled ARIES Targets.

As noted previously, ARIES generates two types of reply environments: (1) modeled replies generated by the three MAT generators as commanded by the ARIES computer via the MRC just described, and (2) fruit replies generated by the four FAT generators using control inputs originating in the ARPG and the MRPG as controlled by the FRC. The type of reply that drives these identical MAT and FAT modules is determined by the controller. These modules are referred to as "reply generators" in the following sections.

The reply generators accept 4- or 10-word message blocks (4 words for an ATCRBS reply and 10 words for a Mode S reply). Each reply generator module has buffer space for 5 ATCRBS or 2 Mode S replies so that a consecutive string of replies can follow one another as rapidly as possible without controller intervention. Each message block contains the following data fields:

a. Trigger Time (range)

b. Reply Code for ATCRBS; Message block for Mode S

c. Power Level

d. Monopulse Off-boresight Angle

e. Mainbeam or Sidelobe Reply (MAT replies are always mainbeam; FAT replies occur randomly as mainbeam or sidelobe replies determined by the fruit environment scenario)

Reply generator action is initiated when the 20-bit trigger-time word is coincident with a 16-MHz reply trigger-time counter. The target generator modulator then produces a PAM data stream for ATCRBS replies and a pulse position modulation (PPM) data stream for Mode S replies. The modulating waveform, either ATCRBS or Mode S, plus the power level, monopulse offboresight angle, and mainbeam/sidelobe control signals are sent to the reply generator IF unit which produces the actual reply at 60 MHz.

6.4.1.2.1 Reply Generator Data Bus Structure.

Each reply generator receives its data over the Y-Bus and its commands over the C-Bus from the controller. The generator reports its status, i.e., buffer full condition, to the controller using the D-Bus. As previously shown in figure 6.4.1-1, all of the reply generators share the same buses and each generator can receive (send) data from (to) the bus only if the control commands on the C-Bus are addressed to it. The device address and OP codes used to form the eight control commands are shown in figure 6.4.1.2.1-1; definitions of these commands are as follows:

a. READ STATUS — Read reply generator status.

b. RESET — Initialize reply generator.

c. LOAD 4 — Load four words of an ATCRBS reply, or the first four words of a Mode S reply, into the buffer.

COMMAND	DEVICE ADDRESS				OP CODE				
READ STATUS	0	0	0	0	0	0	0	0	
RESET	0	0	0	0	0	0	0	1	
LOAD 4	0	0	0	0	0	0	1	0	
LOAD 6	0	0	0	0	0	0	1	1	REPLY
READ 4	0	0	0	0	0	1	0	0	GENERATOR #1
CH - ON	0	0	0	0	0	1	0	1	
CH - OFF	0	0	0	0	0	1	1	0	
DIAGNOSTIC MODE	0	0	0	0	0	1	1	1	
COMMAND BITS (MSB)	0	1	2	3	4	5	6	7	(LSB)

COMMAND	DEVICE ADDRESS				OP CODE				
READ STATUS	0	0	0	1	0	0	0	0	
RESET	0	0	0	1	0	0	0	1	
LOAD 4	0	0	0	1	0	0	1	0	
LOAD 6	0	0	0	1	0	0	1	1	REPLY
READ 4	0	0	0	1	0	1	0	0	GENERATOR #2
CH ON	0	0	0	1	0	1	0	1	
CH - OFF	0	0	0	1	0	1	1	0	
DIAGNOSTIC MODE	0	0	0	1	0	1	1	1	
COMMAND BITS (MSB)	0	1	2	3	4	5	6	7	(LSB)

COMMAND	DEVICE ADDRESS				OP CODE				
READ STATUS	0	0	1	0	0	0	0	0	
RESET	0	0	1	0	0	0	0	1	
LOAD 4	0	0	1	0	0	0	1	0	
LOAD 6	0	0	1	0	0	0	1	1	REPLY
READ 4	0	0	1	0	0	1	0	0	GENERATOR #3
CH - ON	0	0	1	0	0	1	0	1	
CH - OFF	0	0	1	0	0	1	1	0	
DIAGNOSTIC MODE	0	0	1	0	0	1	1	1	
COMMAND BITS (MSB)	0	1	2	3	4	5	6	7	(LSB)

COMMAND	DEVICE ADDRESS				OP CODE				
READ STATUS	0	0	1	1	0	0	0	0	
RESET	0	0	1	1	0	0	0	1	
LOAD 4	0	0	1	1	0	0	1	0	
LOAD 6	0	0	1	1	0	0	1	1	REPLY
READ 4	0	0	1	1	0	1	0	0	GENERATOR #4
CH - ON	0	0	1	1	0	1	0	1	
CH - OFF	0	0	1	1	0	1	1	0	
DIAGNOSTIC MODE	0	0	1	1	0	1	1	1	
COMMAND BITS (MSB)	0	1	2	3	4	5	6	7	(LSB)

FIGURE 6.4.1.2.1-1. DEVICE ADDRESS AND OP CODE FOR REPLY GENERATORS

d.	LOAD 6	Load the remaining six words of a Mode S reply.
e.	READ 4	Read four words of a transmitted ATCRBS reply when in diagnostic mode.
f.	CW ON	Set continuous wave (CW) modulation.
g.	CW OFF	Clear continuous wave modulation.
h.	DIAGNOSTIC	Place in diagnostic mode of operation.

Following is the normal sequence of reply loading. The controller sends out a read status command to the reply generator. Depending on the status of the input buffer, the reply generator will respond with the buffer status bit set to "1" buffer not full or "0" for buffer full. When a "1" is received by the controller, it will send a LOAD 4 command followed by four reply words. If the reply is a Mode S reply type, the controller will then send a LOAD 6 command and output the remaining six words of the Mode S reply block. The controller then polls the next reply generator likewise. The controller will continue this round-robin process as long as reply data are available from the computer.

The reply generators are reset by the RESET command issued by the MRC.

6.4.1.2.2 Major Elements of the Reply Generator.

Figure 6.4.1.2.2-1 shows the mayor elements of the reply generator which consists of:

a. Input Buffer and Registers
b. Delay-to-Trigger Timing Circuitry
c. Reply Assembler

The reply generator receives two types of data from the controller: (1) the reply data to be included in the ATCRBS and Mode S reply, and (2) the reply control data. The reply control data consists of: reply time, power level of the reply, offboresight angle of the reply, and the M/S control bit.

The reply blocks are stored in the Input Buffer and Registers. They are then routed so that the reply data are forwarded to the Reply Assembler, the reply time to the Delay-to-Trigger Timing Circuitry, the power level, the offboresight angle, and the M/S bit to the IF Unit.

The reply data are transferred to the Reply Assembler 16 bits at a time. The assembler is then pulsed serially to control the PAM/PPM modulation, thus simulating the actual ATCRBS/Mode S reply pulse train. The ATCRBS reply data is contained in one 16-bit word. The Mode S reply data are contained in three and one-half words, or seven words, depending whether the reply is a short or long type, respectively.

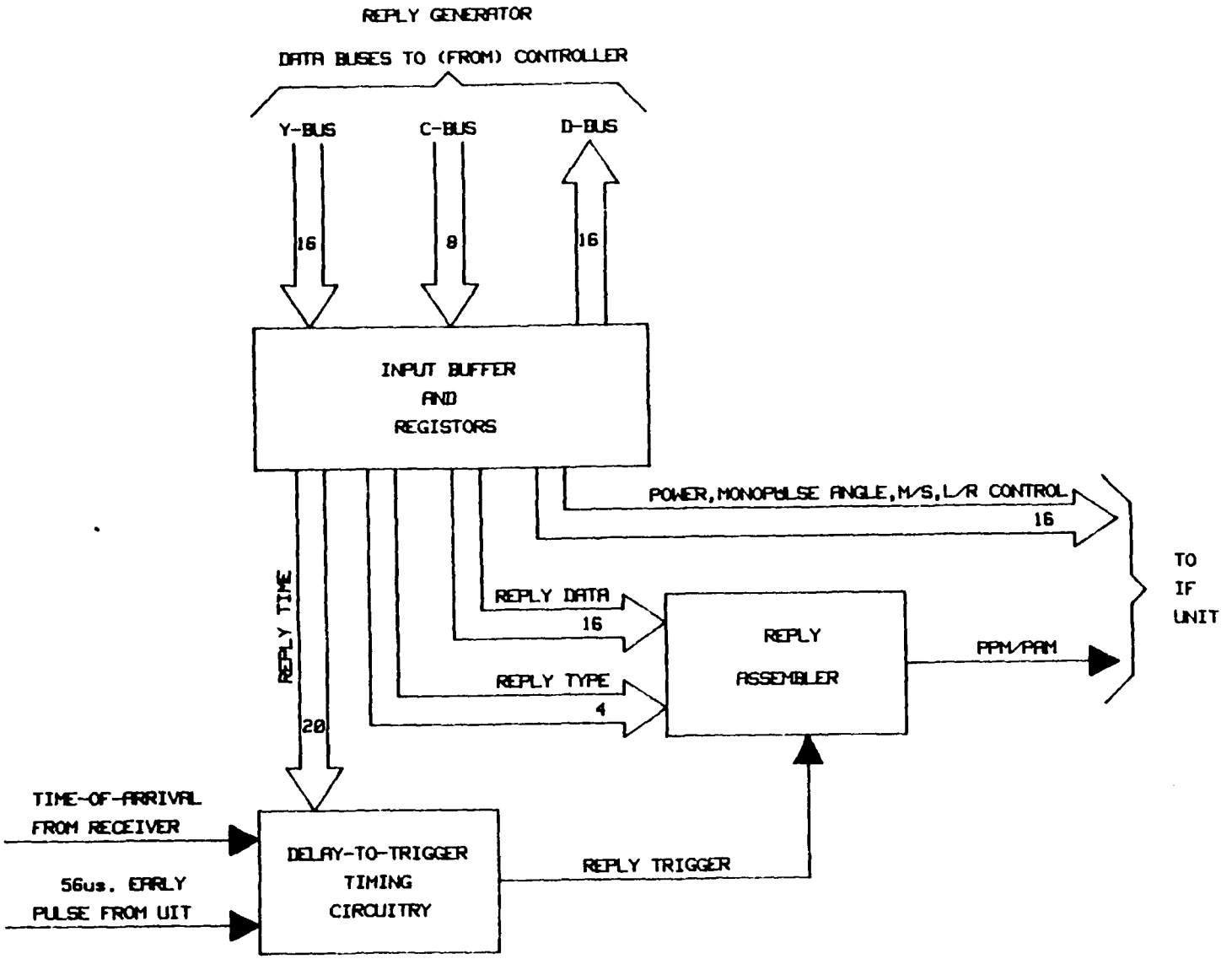


FIGURE 6.4.1.2.2-1. REPLY GENERATOR BLOCK DIAGRAM

To simulate the range of the reply, the reply time is compared with a 20-bit range clock in the Delay-to-Trigger Timing Circuit. When a comparator indicates that coincidence has occurred, a reply trigger is generated causing the Reply Assembler to transmit the modulated reply. The other three control words of the reply are sent to the IF Unit. These remain effective during the entire reply transmission period so that the required characteristics of the reply are simulated.

6.4.1.2.2.1 Input Buffer and Registers.

A detailed block diagram of the reply generator is shown in figure 6.4.1.2.2.1-1. Note that the Input Buffer and Registers consists of:

a. 17-bit x 16-word input buffer and its read/write address registers
b. Hold-1 register and Hold-1 flip-flop
c. Hold-2 register and Hold-2 flip-flop
d. C-Bus Decoder
e. State Controllers No. 1 and No. 2

The 17-bit x 16-word buffer is for the purpose of stacking replies so that they can follow one another without controller intervention. Before the reply data are sent to the Reply Assembler and control data to the IF Unit, output from the buffer is further buffered by the Hold-1 and Hold-2 registers. (The purpose for these two registers will become apparent later in the discussion.) To indicate the status of the Hold-1 register, the Hold-1 flip-flop is set (or reset) each time a new set of data words are transferred to (from) the register. The Hold-2 flip-flop performs similarly for the Hold-2 register. The writing of new data into the buffer and the transferring of data from the buffer to the Hold-1 register are controlled by State Controller No. 1. Transferral of data from the Hold-1 register to the Hold-2 register is controlled by State Controller No. 2. State diagrams for the two state controllers are shown in figures 6.4.1.2.2.1-2 and 6.4.1.2.2.1-3.

Consider the following Input Buffer and Register action sequence occurring during transfer of data from the controller (power up condition is assumed to have just occurred):

a. The reply generator (MAT) is reset by the MRC via a RESET command. The command is decoded by the bus decoding logic monitoring the C-Bus as shown in figure 6.4.1.2.2.1-4. The reset causes the read address register and write address register of the buffer to return to 0, the Hold-1 and Hold-2 flip-flops to be cleared, and State Controller No. 1 and No. 2 to be reset to their idle states. The reply generator will remain in the idle state as long as no reply blocks are received.

b. When the MRC is ready to send reply blocks, it will first check the status of the buffer by sending READ STATUS command. This command is decoded by the bus decoder that was addressed which in turn responds by placing the reply generator status word on the D-BUS. This word is checked by the MRC to determine whether the input buffer is full (Busy).

c. The Busy ROM determines the status of the buffer by comparing the 4-bit outputs of the Read Address Register (RAR) and Write Address Register (WAR). If there are nine or fewer locations left for writing, a busy bit from the ROM will be set indicating "buffer full" status. The buffer can thus accept at least one full 10-word Mode S reply when it is not full.

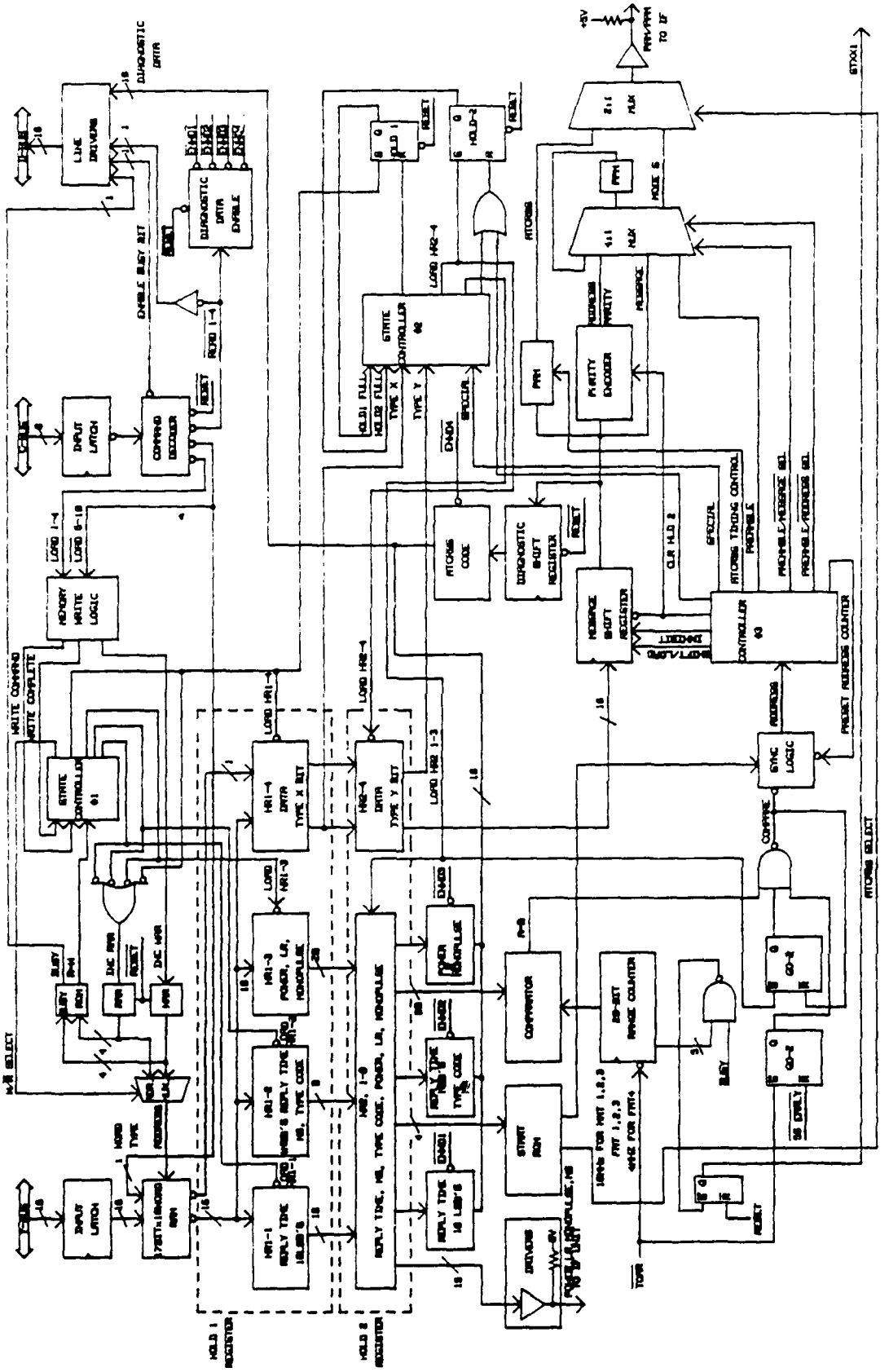


FIGURE 6.4.1.2.2.1-1. REPLY GENERATOR DETAILED BLOCK DIAGRAM

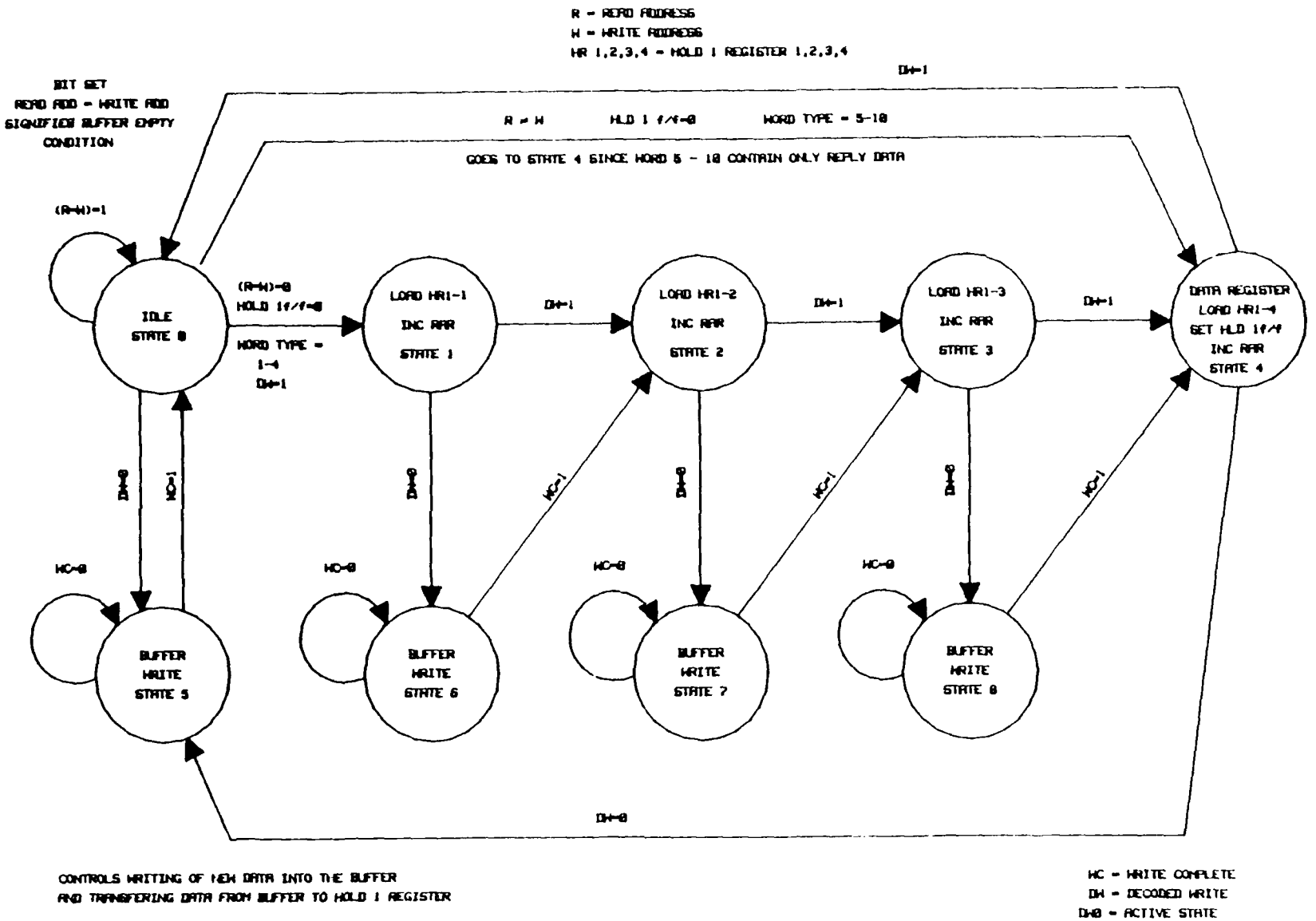


FIGURE 6.4.1.2.2.1-2. STATE DIAGRAM FOR STATE CONTROLLER No. 1

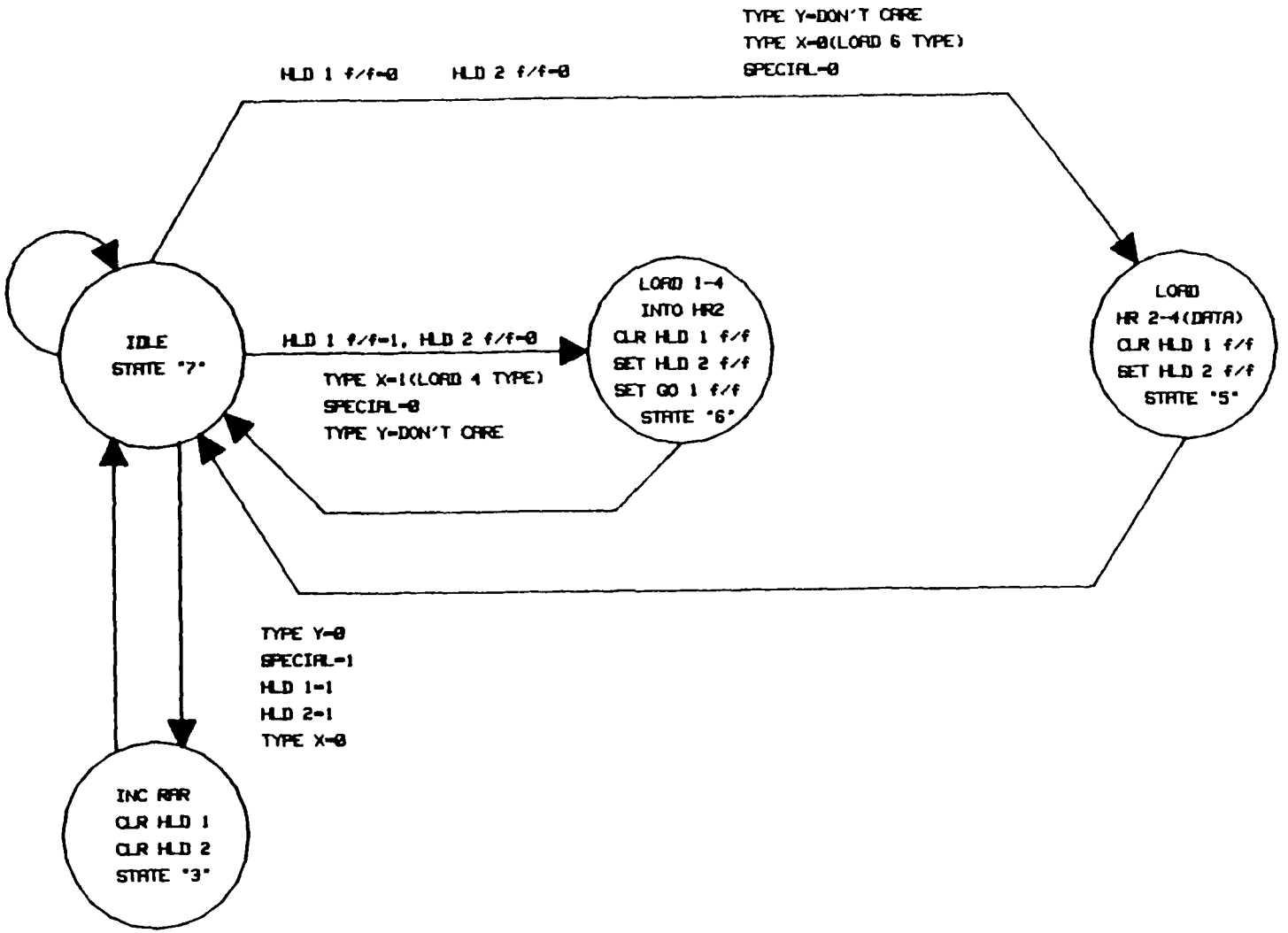


FIGURE 6.4.1.2.2.1-3. STATE DIAGRAM FOR STATE CONTROLLER No. 2

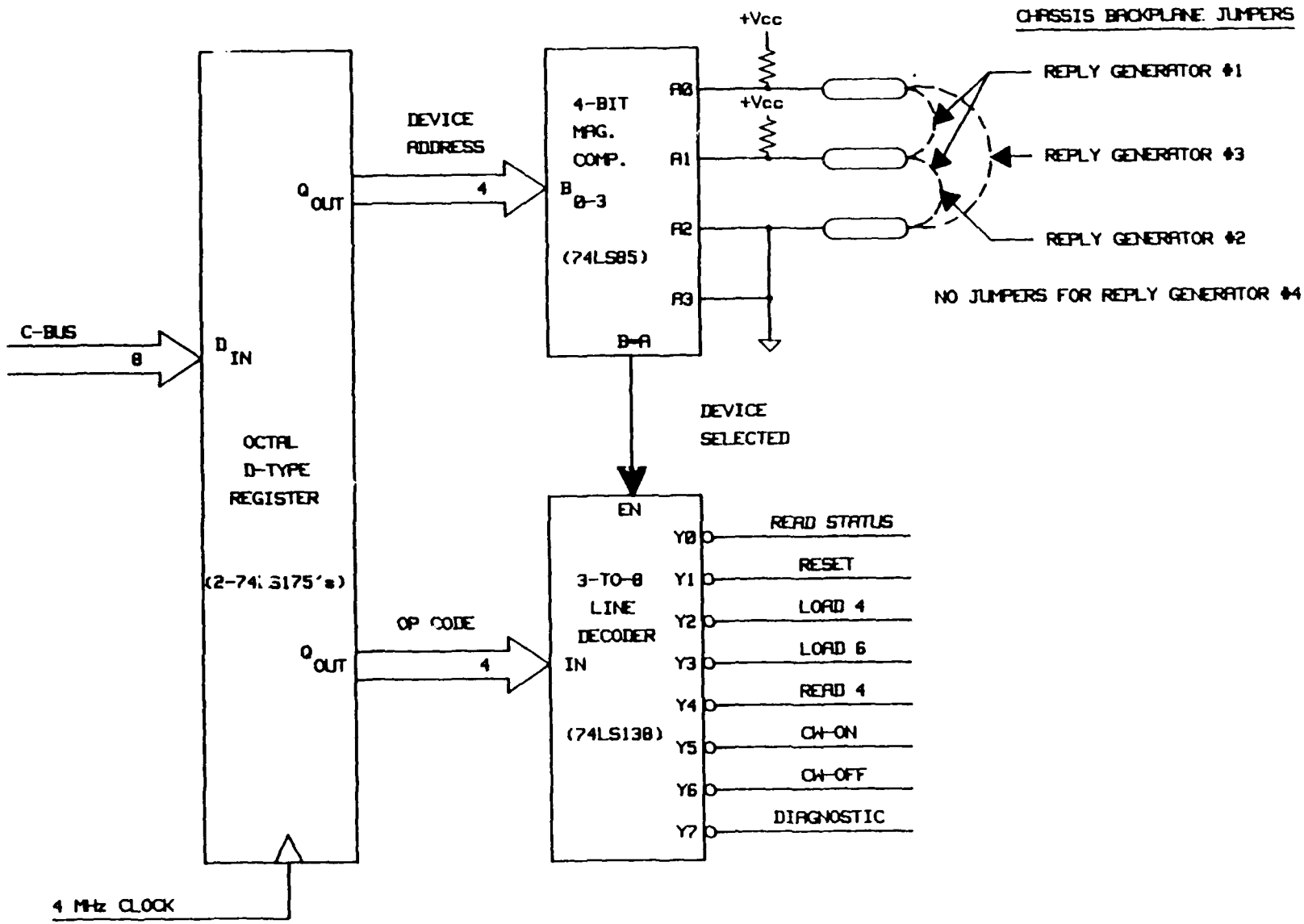


FIGURE 6.4.1.2.2.1-4. REPLY GENERATOR CONTROL BUS DECODING CIRCUITRY

d. Another bit from the Busy ROM signifies the "buffer empty" condition. It is set when the read address is equal to the write address. This R=W bit is used by State Controller No. 1 for reasons to be explained below.

e. Since the reply generator has just been powered up, the R=W bit is set. This condition causes State Controller No. 1 to loop around the Idle State (state 0) until a write command from the controller is received (see figure 6.4.1.2.2.1-2).

f. It is important at this point to notice how the ATCRBS and Mode S replies are transferred. An ATCRBS reply is transferred by a LOAD 4 command accompanied by four data words. A Mode S reply is transferred in a similar way using a LOAD 4 command accompanied by the first four words of the reply block, followed by using a LOAD 6 command accompanied with the remaining six words of the reply block.

g. When a LOAD command is decoded by the Bus Decoder, a "Write" pulse will be generated by the Buffer Write Logic to State Controller No. 1, causing four words to be written into the buffer if it is a LOAD 4 command and six words, if it is a LOAD 6 command. Figure 6.4.1.2.2.1-5 illustrates the Buffer Write Logic and figure 6.4.1.2.2.1-6 illustrates the waveform for the LOAD 4 and LOAD 6 sequence.

h. At the completion of writing, a "Write Complete" pulse from the Buffer Write Logic will cause State Controller No. 1 to return to state 0 from state 5 as illustrated on the state diagram in figure 6.4.1.2.2.1-2. Since the write address has been incremented, the read address is no longer equal to the write address. Also, at this time the Hold-1 flip-flop is in its 0 state as the Hold-1 register is empty, and the transfer was a LOAD 4 type. These three conditions will cause the received four words in the buffer to be loaded into the "reply time" register (HR1-1), "power, monopulse, left/right (L/R)" register (HR1-2), "reply type" register (HR1-3), and the "DATA" register (HR1-4) of the Hold-1 register array, as indicated by states 1, 2, 3, and 4, respectively. At the end of state 4, the Hold-1 flip-flop will be set signifying a "Hold-1 register full" condition, and state 0 is reentered. Note that during states 0, 1, 2, 3, and 4, if a "write" pulse is decoded, State Controller No. 1 will process the buffer write sequence before going into the next state. Also, note that state 0 can go directly to state 4 if the received data is from the LOAD 6 command. This is due to the fact that word 5 through 10 of the Mode S reply block contains only the reply data. State 4 will enable these words to be loaded into the "DATA" register so that the reply data will be encoded and assembled by the Reply Assembler (see Hold-1 register in figure 6.4.1.2.2.1-1).

There are three conditions that must be met before the reply words can be transferred from Hold-1 register to Hold-2 register. These conditions are:

a. Hold-1 register is full (Hold-1 flip-flop = "1".)

b Hold-2 register is empty (Hold-2 flip-flop = "0".)

c. The Special Bit is "0." (The special bit is generated by State Controller No. 3 at the end of the reply.)

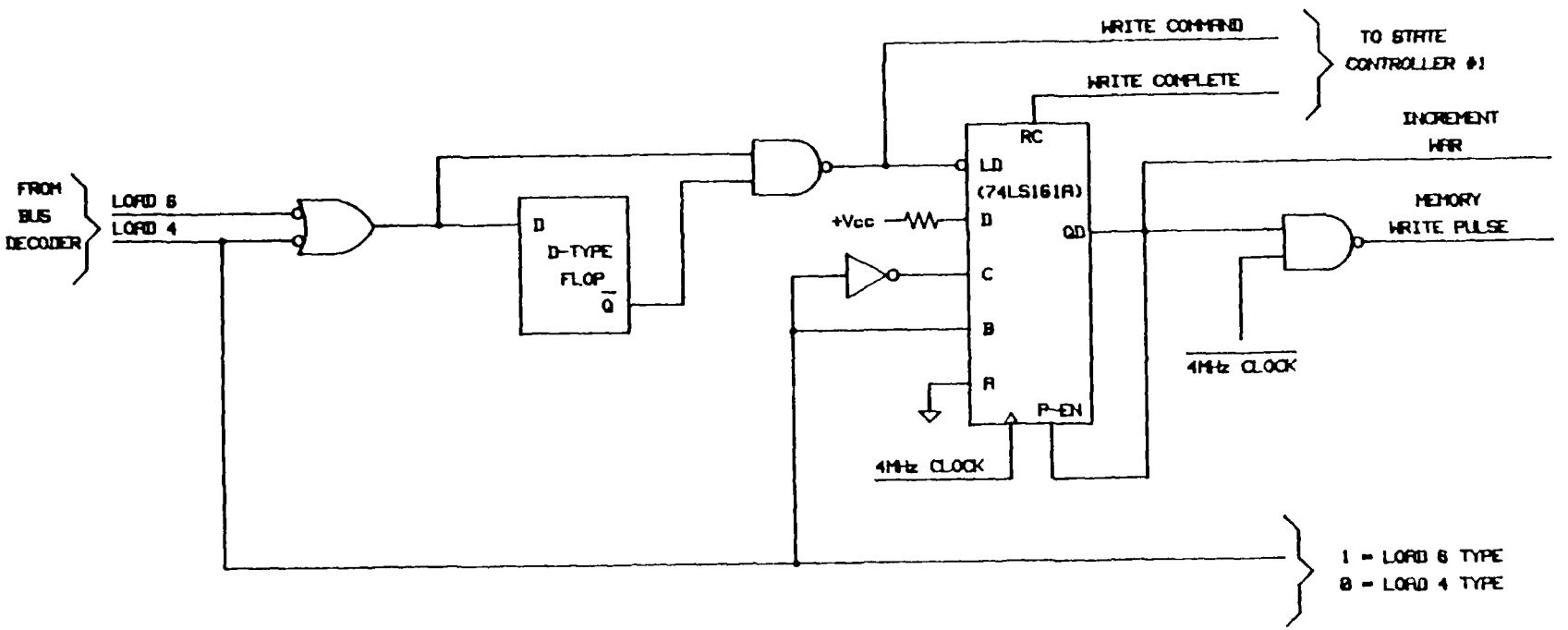


FIGURE 6.4.1.2.2.1-5. BUFFER WRITE LOGIC

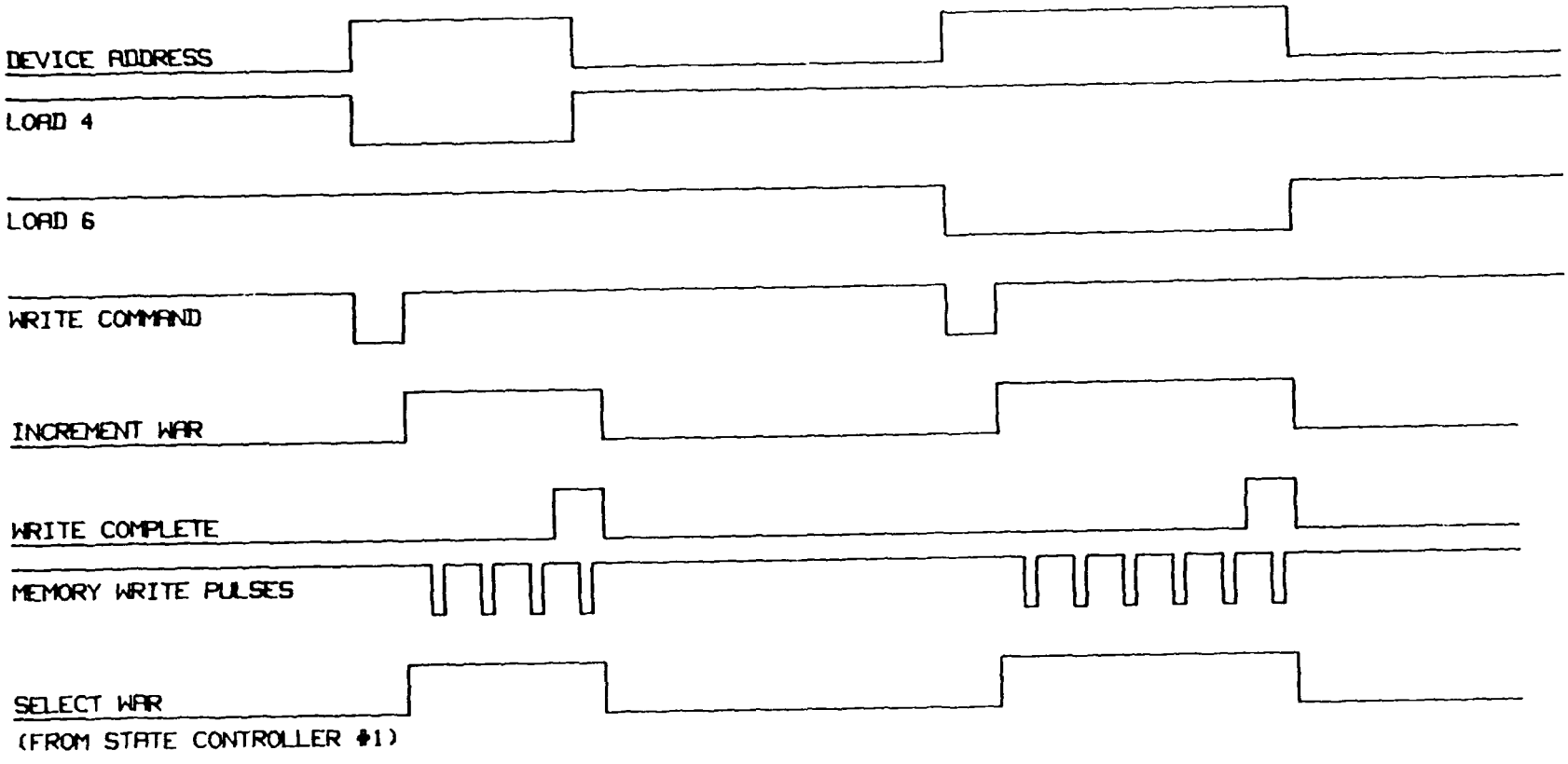


FIGURE 6.4.1.2.2.1-6. BUFFER WRITE TIMING WAVEFORMS

Two other bits also affect the transfer. They are the type X and type Y bits. A type X or type Y bit is an additional bit attached to each word of the reply. Its sole purpose is to indicate whether the reply word is a LOAD 4 type or LOAD 6 type as discussed previously. Type X comes from the Hold-1 register, and when it is transferred to the Hold-2 register, it becomes a type Y.

Referring to figure 6.4.1.2.2.1-3, if the type X bit is equal to a 1, indicating that the word in Hold-1 register is a LOAD 4 type, State Controller No. 2 will cause the contents of the entire Hold-1 register to be parallel loaded into the Hold-2 register (state 6). On the other hand, if type X is equal to a 0, indicating a LOAD 6 type, only the contents of the DATA register will be transferred (state 5). In this way, after a LOAD 4 transfer, the Hold-2 register will contain the reply control words and a reply data word. If the reply is a Mode S type, the remaining six words of the reply data will be transferred as a LOAD 6 type. During this transfer, only the DATA register is affected. The control words will remain unchanged in the Hold-2 register during the entire reply transmission.

Each time data of either type is transferred to Hold-2 register, the Hold-2 flip-flop will be set and the Hold-1 flip-flop will be cleared by State Controller No. 2.

Resetting the Hold-1 flip-flop will cause additional data, if any, stored in the buffer, to be immediately transferred into the Hold-1 register and subsequently, into the Hold-2 register when the Hold-2 flip-flop is cleared.

The purpose of the two-stage buffering done by the Hold-1 and Hold-2 registers is to enable replies to follow one another with *minimum separation in time*. This is effected by having the Hold-2 register and the Reply Assembler processing the last reply data word for the current reply concurrently with the next reply being loaded into the Hold-1 register. When the current reply is terminated, the next reply is loaded into the Hold-2 register and Reply Assembler for immediate processing.

The ARIES can simulate the following four types of Mode S replies: (1) surveillance, (2) All-Calls, (3) Comm-B, and (4) Comm-D. The latter two are 112 bits long as compared to 56 bits for the other two. For simplicity, all Mode S replies are transferred to the reply generator in a 10-word block. Therefore, there will be three "don't care" words in the All-Call and surveillance replies. (See appendix D of the ARIES Hardware Maintenance Manual for reply block formats.) The reply generator ignores these three words in the two types of short replies. This is done by state 3 of State Controller No. 2 as depicted in figure 6.4.1.2.2.1-3. To understand how this is accomplished, assume that the Reply Assembler is processing the 7th (last) word of a short reply, and that therefore, the 8th, 9th, and 10th word of the same reply block are stored in the Hold-2 register, Hold-1 register, and the current read location of the buffer, respectively. Normally, at the end of transmission of each word, the Hold-2 flip-flop is cleared by State Controller No. 3. This allows the next data word to be propagated to the Hold-2 register. However, at the end of the 7th word of a short reply, the Hold-2 flip-flop is not cleared, therefore the Hold-1 and Hold-2 flip-flops remain set and the type X and type Y bits are 0 since the 8th and 9th words are LOAD 6 type. These conditions signify that the current reply is finally transmitted, as indicated by the Special Bit = 1, state 3 will be entered, causing the Hold-1 and Hold-2 flip-flops to be cleared, and the current read location of the buffer to be skipped.

6.4.1.2.2.2 Delay-to-Trigger Timing Circuit.

The Delay-to-Trigger Timing Circuit consists of:

a. 20-bit range counter
b. 20-bit magnitude comparator
c. GO-1 and GO-2 flip-flops

The first control word and the upper nibble of the second control word in a reply block, either ATCRBS or Mode S, is the 20-bit reply time. It specifies the precise moment that a reply should be transmitted by the Reply Assembler. The reply time is held in the Hold-2 register, along with other control words, and is continuously compared to the 20-bit range clock in the Delay-to-Trigger Timing circuit. When the value of the reply time and range clock are equal, a trigger pulse is generated by the comparator which causes the reply to be transmitted.

The GO-1 and GO-2 flip-flops must be set before a trigger pulse can be forwarded to the Reply Assembler. The reason for these two flip-flops are discussed below.

Since the comparator continuously compares the contents of the range clock and the reply time register, whether reply data is present or not, unwanted trigger pulses may be generated. To prevent this from happening, gating logic is required. This is the purpose of the GO-1 flip-flop. It is set whenever a LOAD 4 type is loaded into the Hold-2 register, and reset when the compare pulse is generated. The GO-1 flip-flop remains reset until another LOAD 4 type is loaded.

There is a period of at least 56 μs (dead time) between the beginning of the last Mode S reply and the next ATCRBS/Mode S All-Call interrogation when no reply should be generated. A Mode S reply generated in this 56-μs interval would overlap the interrogation, and the interrogation scheduling software prevents this.

During this period, ATCRBS replies for the upcoming interrogation are being loaded into the reply generator. Following the usual loading sequence, the reply may be in the Hold-2 register waiting to be transmitted. If the reply time happens to be equal to the range counter at this time, the reply would be triggered prematurely before the arrival of the All-Call interrogation. The GO-2 flip-flop prevents this from happening. A 56 EARLY pulse is generated by the UIT 56 μs before the next All-Call interrogation causing the GO-2 flip-flop to be reset; and thus preventing a compare pulse from being sent. When the All-Call interrogation is finally decoded by the Uplink Receiver, the TOAR pulse from the receiver will set the GO-2 flip-flop. Also, the TOAR resets the range counter back to 0, synchronizing the reply generator with the Uplink Receiver.

6.4.1.2.2.3 Reply Assembler.

The Reply Assembler is the heart of the reply generator. It takes reply data from the "data register" of the Hold-2 register and converts it to serial pulses duplicating the PAM/PPM modulation format of the ATCRBS/Mode S replies.

The Reply Assembler consists of:

a. A shift register that performs parallel to serial conversion
b. A parity encoder for Mode S replies
c. PAM modulator for ATCRBS replies

d. PPM modulator for Mode S replies
e. Synchronization logic for timing
f. State Controller No. 3 for controlling the Reply Assembler
g. Multiplexers for selecting Mode S and ATCRBS replies

State Controller No. 3 consists of a 24-bit x 512-word ROM, and a multiplexing circuit shown in figure 6.4.1.2.2.3-1. The ROM is coded to generate all necessary signals to control the data shift register, PPM modulator, and PAM modulator. The control signals for each type of reply (ATCRBS, Long Mode S, and Short Mode S) are available from the ROM at all times. The multiplexing circuits select the correct type for the current reply. The multiplexing circuits are controlled by the least significant four bits of the third control word of a reply block. (See appendix D of the ARIES Hardware Maintenance Manual for these formats.)

To generate a reply, the Reply Assembler must first receive a trigger pulse from the Delay-to-Trigger Timing Circuitry. Depending on the reply type stored in the Hold-2 register, the trigger pulse will preset the Sync logic to a starting address for the State Controller No. 3 ROM. All replies, regardless of type, will terminate at location 512 of the ROM (last location). Only the starting addresses are different as each type of reply (ATCRBS, Long Mode S, and Short Mode S) has a different time period. Table 6.4.1.2.2.3-1 shows the starting address for each reply and figure 6.4.1.2.2.3-2 shows the Sync logic. Note that the Sync logic disables itself after the end of each reply and can start only after a trigger pulse (Compare pulse) is received.

TABLE 6.4.1.2.2.3-1. STATE CONTROLLER NO. 3 START ADDRESS TABLE

REPLY TYPE	TYPE 3	TYPE 2	CODE 1	CODE 0	SPI	REPLY TIME (μs)	START ENCODER ROM (Hex)	CONTROLLER NO. 3 START ADDR. (Hex)
ATCRBS w/NO SPI	0	0	0	1	0	21.5	02	1A9
ATCRBS w/SPI	0	0	0	1	1	26.0	03	197
Mode S All-Call	0	0	1	0	X	66.5	04/05	0F5
Mode S Surv.	0	1	0	0	X	66.5	08/09	0F5
Mode S Comm B	1	0	0	0	X	122.5	10/11	015
Mode S Comm D	1	1	0	0	X	122.5	18/19	015

6.4.1.2.3 Reply Simulation.

As mentioned previously, ARIES simulates two type of beacon transponders, ATCRBS, and Mode S. To understand how the Mode S transponder is simulated, consider the following sequences. Assume that the reply to be transmitted is a surveillance type (short) specified by the reply type field held in Hold-2 register. The multiplexer will be selected for the Mode S reply as shown in figure 6.4.1.2.2.1-1. As soon as a trigger pulse is received, Sync Logic will preset State Controller No. 3 ROM address to $0F5_{HEX}$. State Controller No. 3 start addresses are presented in table 6.4.1.2.2.3-1.

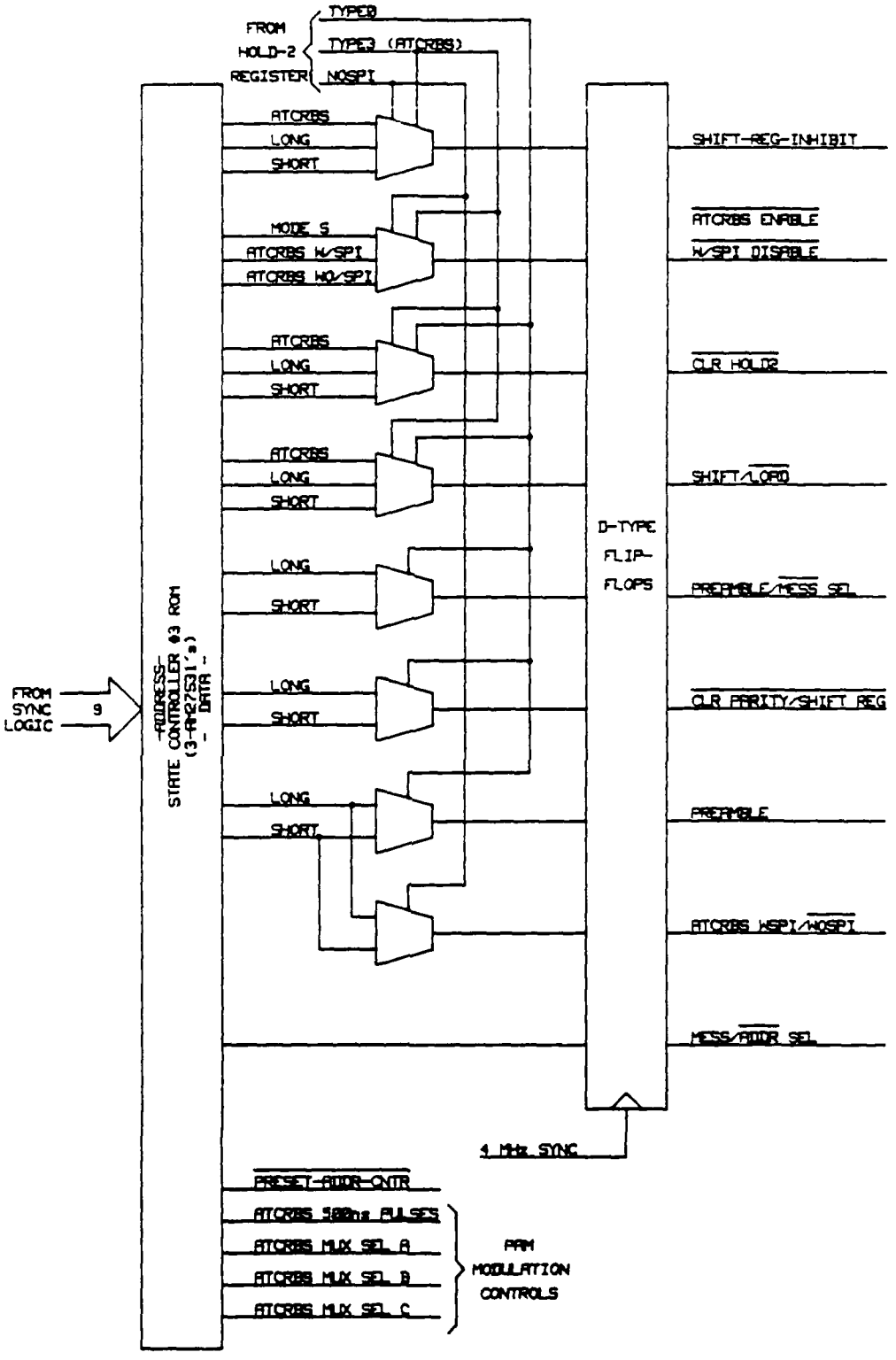


FIGURE 6.4.1.2.2.3-1. STATE CONTROLLER No. 3

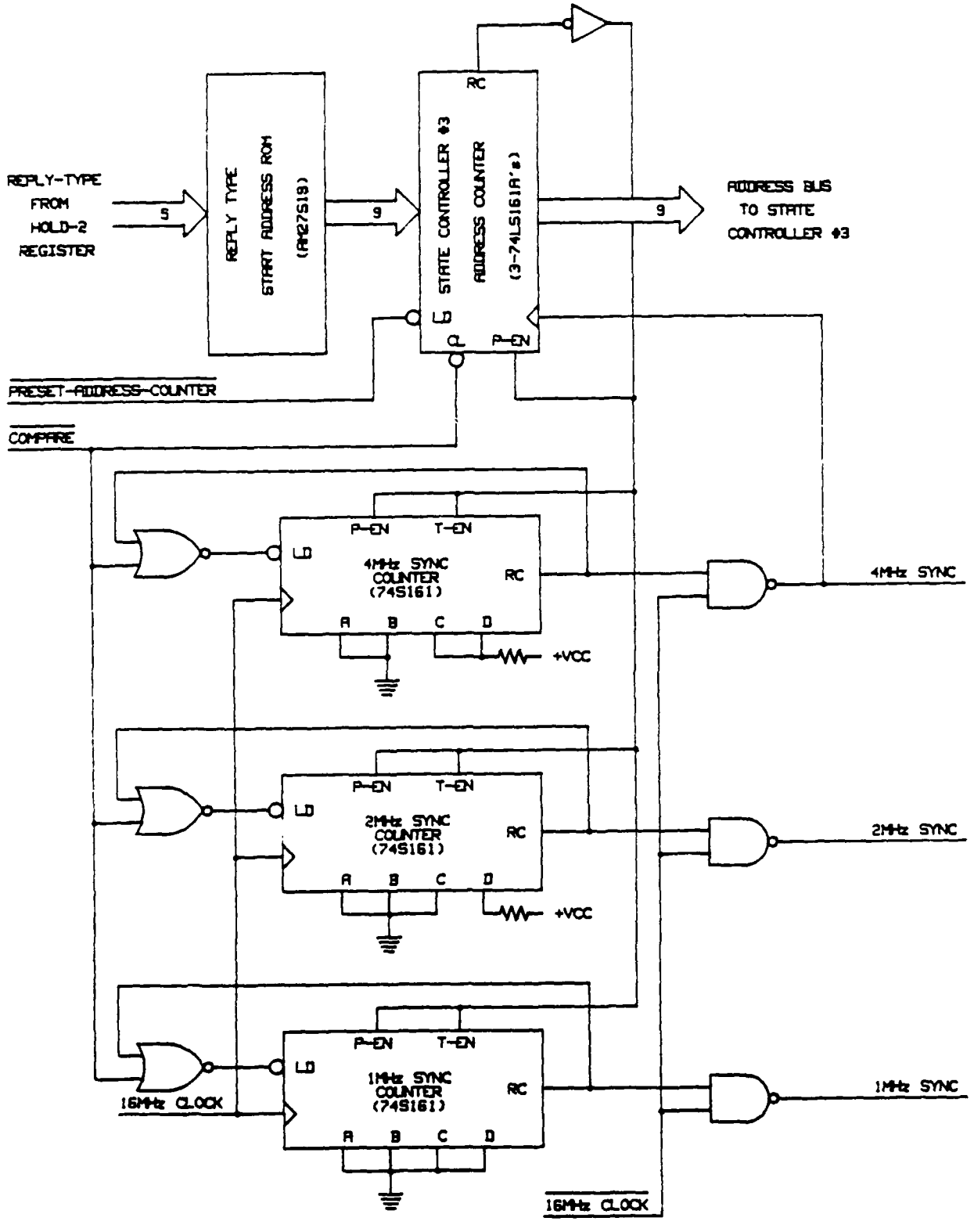


FIGURE 6.4.1.2.2.3-2. SYNC LOGIC CIRCUITRY

The control signal waveforms for the short Mode S reply are depicted in figure 6.4.1.2.3-2. Beginning at address $F5_{Hex}$, the Mode S preamble pulses are the first to be sent, followed by clearing of the shift register and parity encoder register for the upcoming reply. The first 16 bits of the reply data are then loaded into the shift register from the Hold-2 register. The shift register is clocked every 1 μs so that the data are shifted out at a rate corresponding to the Mode S reply data rate (1 megabits per second). Loading and shifting for the shift register are controlled by the Shift/Load control line. Note that every time a new set of 16-bit data is loaded into the shift register, the Hold-2 flip-flop will be cleared by the CLEAR Hold-2 control line, in order that consecutive 16-bit data words may be propagated from the memory to the Hold-1 register and to the Hold-2 register as described previously.

Also shown at the lower.half of figure 6.4.1.2.3-2, are the PPM waveforms corresponding to the sample reply bit as generated by the PPM modulator. The sources of the waveform can be identified in figure 6.4.1.2.3-3.

Each reply generator is designed with a 24-bit parity encoder. Its purpose is to take the message portion of the reply and convert it to 24 parity check bits and to add the remaining 24 address bits to these parity check bits in modulo-two fashion. The 24 parity check bits are computed and stored in the parity encoder register as seen in figure 6.4.1.2.3-4. At the end of the message bits (88 for a Mode S long reply), Multiplexer A will be selected so that the last 24 bits of the reply will be coming directly from the parity encoder rather from the shift register. Multiplexer A is controlled by the message/address control line.

Short Mode S replies, All-Calls, and surveillance are simulated in a similar way, except that State Controller No. 3 ROM is started at location 245, and since there are only 32 message bits, Multiplexer A will output the address from the parity encoder after 32 shifts.

The ATCRBS reply format depicted in figure 6.4.1.2.3-5 consists of 16 pulses: 2 bracket pulses F_1 and F_2, 12 data bits, an X bit, and a SPI bit. The width of each pulse is 450 ns, and pulse separation is 1 μs except between F_2 and SPI which is 3.9 μs.

To simulate an ATCRBS reply, output from the message shift register is modulated by the PAM modulator, and Multiplexer C selects the ATCRBS replies.

To understand how the PAM modulator operates, refer to table 6.4.1.2.3-2. Columns 1, 2, and 3 of the table show each ATCRBS reply pulse and its timing relative to F_1. Since the ARIES 4-MHz system clock period is 250 ns, it is not possible to match the timing of the leading edges of these pulses directly, and a delay scheme must therefore be used. This scheme performs as follows: State Controller No. 3 ROM generates a 500-ns wide pulse with a leading edge that is as close to that of the actual pulse as the resolution of the 4-MHz clock permits, but which does not occur later than the desired leading edge. Leading edge delays for these pulses are shown in column 4. Delay is then added to set the leading edge of the simulated pulse at its proper value. These delays vary from 0 to 200 ns in increments as shown in column 5.

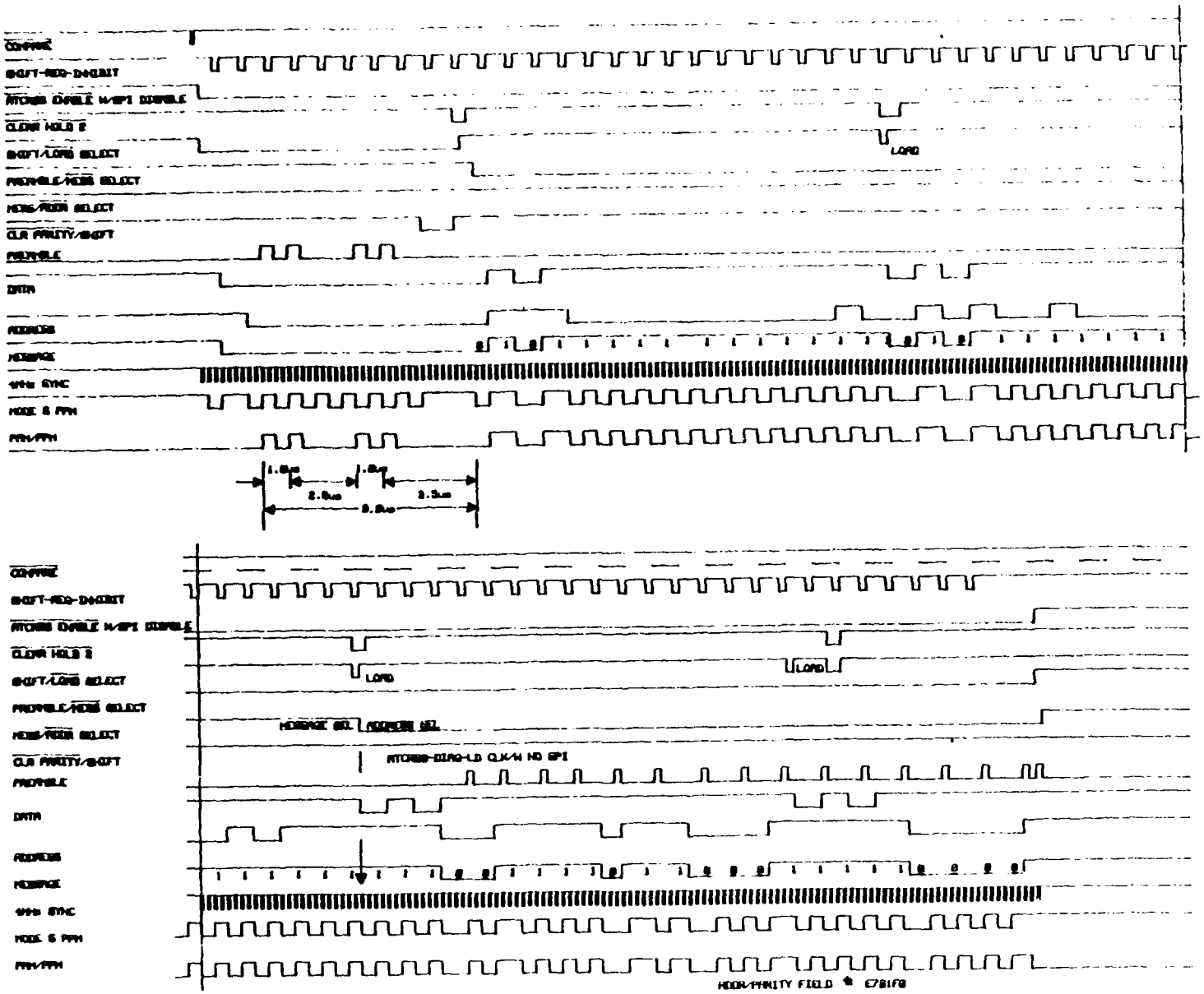


FIGURE 6.4.1.2.3-2. MODE S REPLY (SHORT) CONTROL SIGNAL WAVEFORM

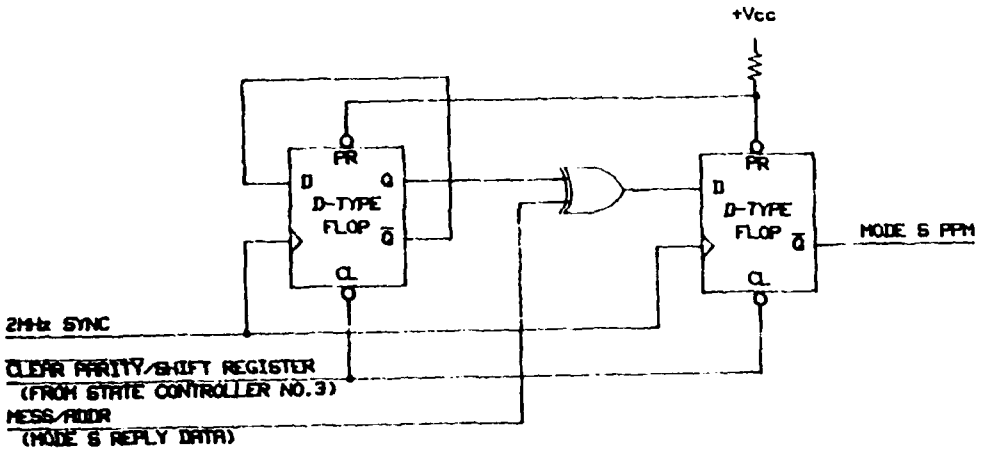


FIGURE 6.4.1.2.3-3. PPM MODULATOR

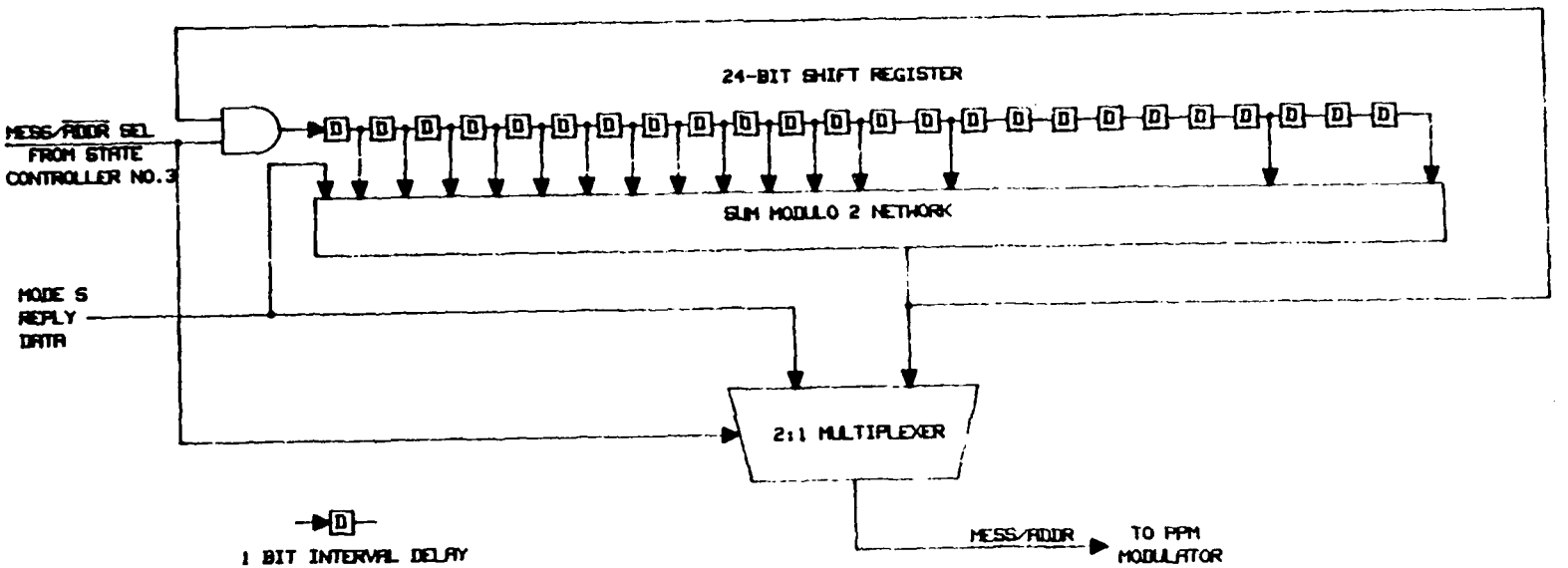


FIGURE 6.4.1.2.3-4. FUNCTIONAL DIAGRAM OF MODE S TRANSPONDER PARITY ENCODER

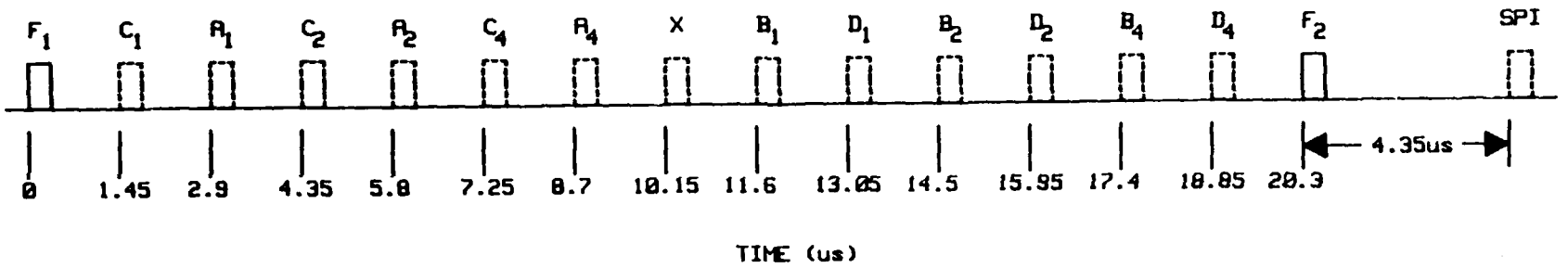


FIGURE 6.4.1.2.3-5. ATCRBS REPLY WAVEFORM

TABLE 6.4.1.2.3-2. TIMING FOR AN ATCRBS REPLY

(1) ATCRBS PULSE	(2) ACTUAL PULSE TIMING	(3) SHIFT REGISTER OUT	(4) CONTROLLER #3 ROM OUT	(5) PROGRAMMABLE DELAY
F_1	0	-.25	0	0
C_1	1.45	1.25	1.25	.200
A_1	2.90	2.50	2.75	.150
C_2	4.35	4.00	4.25	.100
A_2	5.80	5.50	5.75	.050
C_4	7.25	7.00	7.25	0
A_4	8.70	8.50	8.50	.200
X	10.15	9.75	10.00	.150
B_1	11.60	11.25	11.50	.100
D_1	13.05	12.75	13.00	.050
B_2	14.50	14.25	14.50	0
D_2	15.95	15.75	15.75	.200
B_4	17.40	17.00	17.25	.150
D_4	18.85	18.50	18.75	.100
F_2	20.30	20.00	20.25	.050

ALL TIMES ARE IN MICROSECONDS

THE SUM OF THESE TWO VALUES MUST BE EQUAL TO THE VALUE IN COLUMN (2).

The PAM modulation circuitry is shown in figure 6.4.1.2.3-6. Signals A, B, C, and D are supplied by State Controller No. 3 ROM: A is a 500-ns pulse. To generate a 450-ns wide ATCRBS pulse, a 50-ns delay element and an AND gate are used. The pulse is then fed into a tapped delay line so that appropriate delays may be selected by the multiplexer. Selection is controlled by the B, C, and D control lines. Output from the multiplexer is then AND'ed with the ATCRBS reply data from the shift register. Note that the data from the shift register must also be shifted out at an appropriate time so that the modulating pulse from the multiplexer will occur at the center of the data bit. The timing for the shift register is shown in column 3. ATCRBS replies with the SPI pulse, appended after the F_2, are generated in a similar way.

6.4.1.3 Modeled Reply Generator Buffer Interface.

The MRG Buffer Interface serves as a link between the computer and the controller and provides buffer space for the computer to store replies as they are generated. It contains two 16-bit x 1024-word memories, along with associated read/write address counters, reply counters, control logic, multiplexers, and bus driver circuitry. As shown in figure 6.4.1.3-1, these two groups of components are labelled with a suffix No. 1 and No. 2.

At any instant of time, one group of components is designated the primary buffer and the other is designated the secondary buffer (i.e., primary memory, primary reply counter, etc.). These designations are reversible via the 56 EARLY control signal generated by the UIT.

To understand how the MRG Buffer Interface works, it is necessary to understand how ARIES handles Mode S Roll-Call and ATCRBS/Mode S All-Call interrogations. This is described fully in the ARIES Operating Software Manual, but is repeated here for convenience.

The Mode S Roll-Call interrogations are handled as they are received. The Uplink Receiver interrupts the computer, which then processes the interrogation data and generates a reply, which is immediately sent to the MRG. This can be done because the specified Mode S transponder turn-around-time of 128 μs allows sufficient processing time.

In the case of ATCRBS/Mode S All-Call interrogations, however, many replies may respond but the specified transponder turn-around-time is only 3 μs. This does not allow sufficient time for the computer to process these interrogations, during real time, even when considering the minimum 1 nmi range of the target, which provides an additional 12 μs for processing. Fortunately, the time and interrogation mode of the ATCRBS/Mode S All-Call interrogations are predictable, as the sensor follows a fixed interrogation pattern. Therefore, the ARIES system can prepare the complete set of ATCRBS and All-Call replies for a given interrogation in advance of the interrogation actually being sent. When the interrogation is received, these precomputed replies are transmitted and ARIES begins generating the replies for the next ATCRBS/Mode S All-Call interrogation.

The precomputed ATCRBS and All-Call replies are stored in the secondary buffer. These replies will not be transmitted since the MRC can only read from the primary buffer. The buffer primary/secondary status are reversed just prior to the next anticipated ATCRBS/Mode S All-Call interrogation. The ATCRBS and All-Call replies are then in the primary buffer and the MRC immediately begins sending them to the MATs.

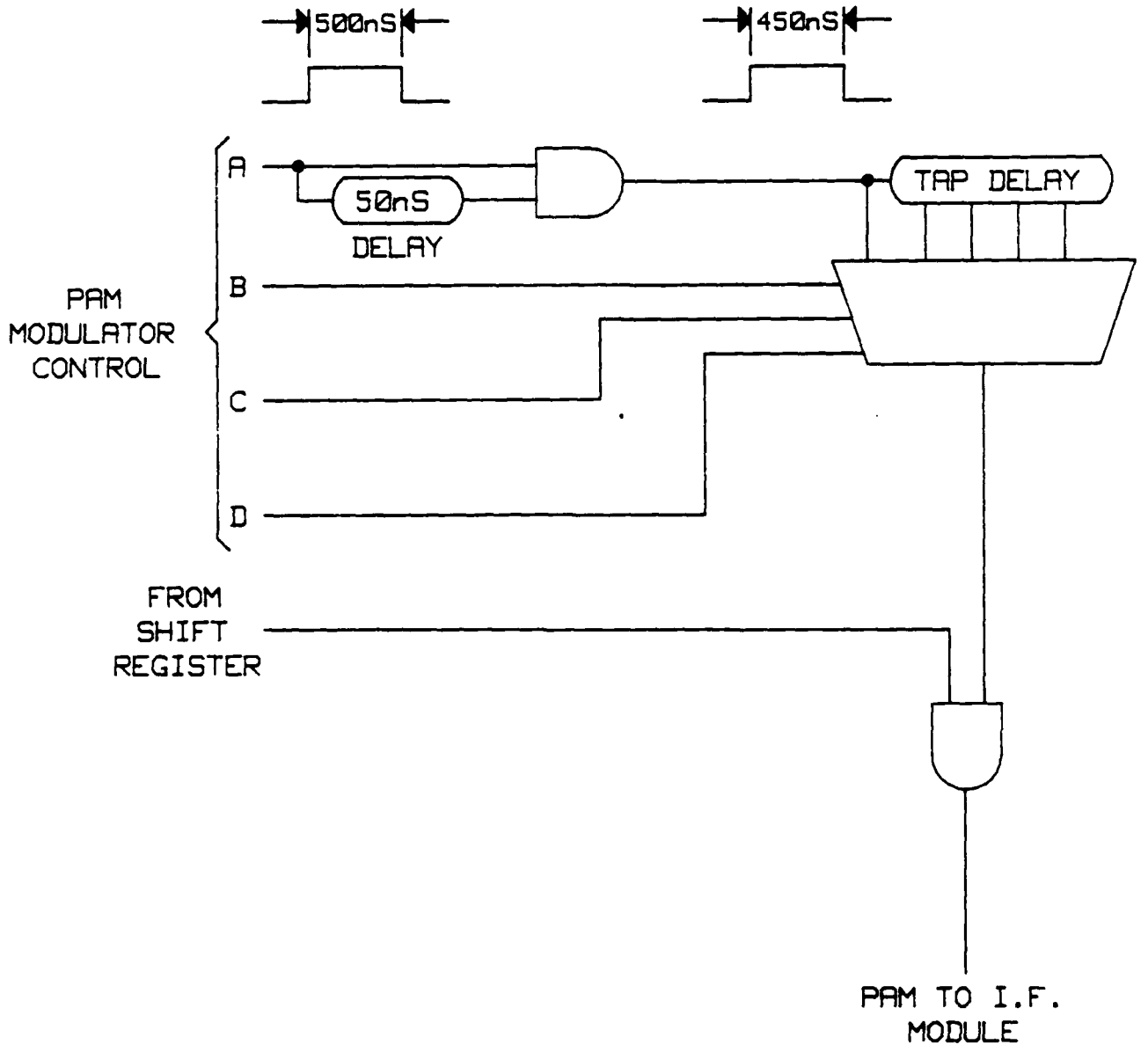


FIGURE 6.4.1.2.3-6. PAM MODULATOR

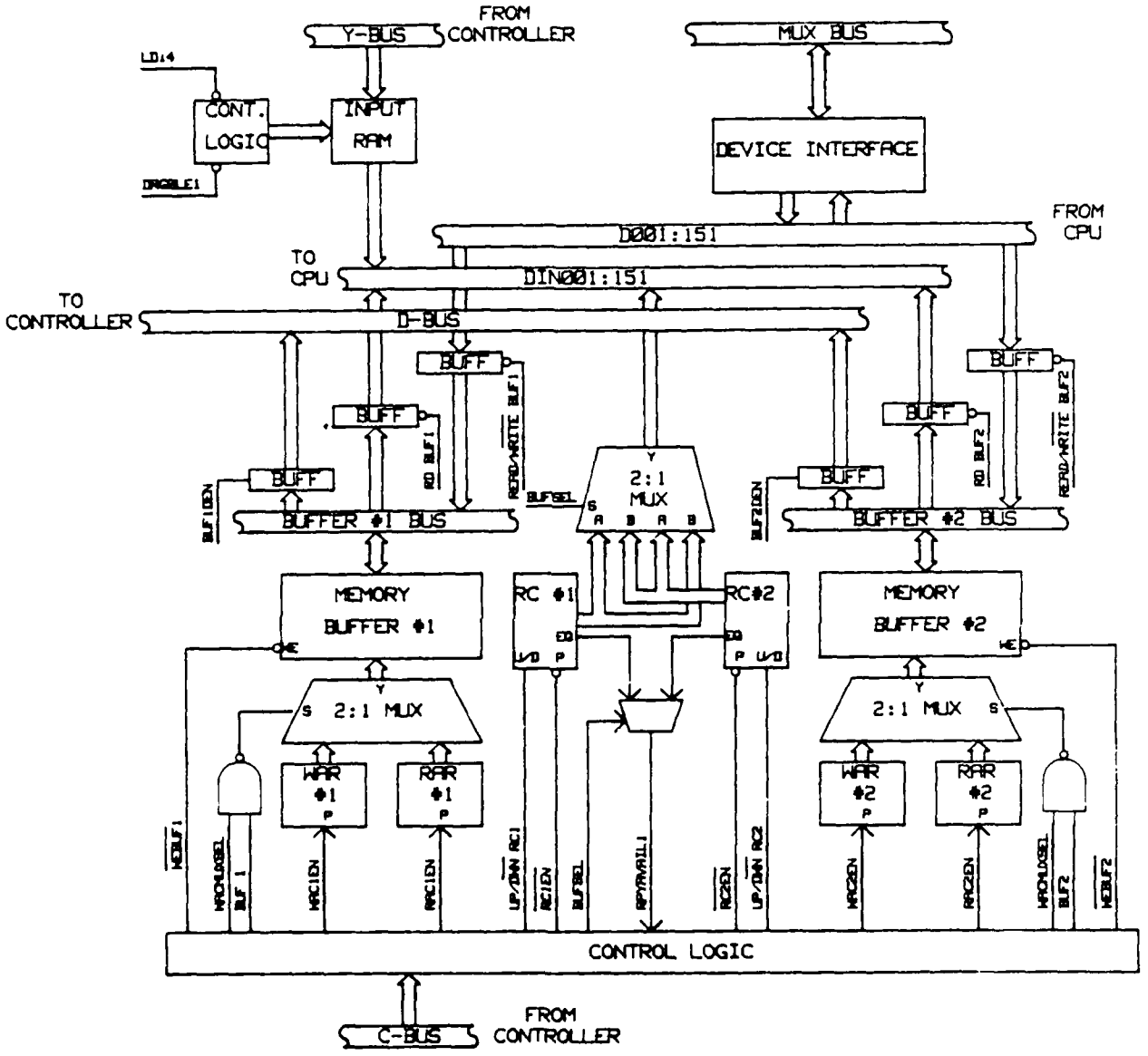


FIGURE 6.4.1.3-1. MRG BUFFER INTERFACE BLOCK DIAGRAM

The Mode S Roll-Call replies are transmitted immediately, and so the computer must store them directly into the primary buffer.

6.4.1.3.1 Memory and Memory Address Logic.

Each memory used in the Interface Buffer consists of four 4-bit x 1024-word memory chips cascaded together to accommodate words, 16-bits wide. Each memory is capable of storing 102 Mode S replies (10 words each) or 256 ATCRBS replies (4 words each). There are separate read and write address counters, so that the memory can be read and written independently. The memory address multiplexer selects whether the read address or the write address is used. All of the address counters are 10-bit circular type. They are reset to 0 by any one of the following conditions: (1) during system power up, (2) under program control via reset command, and (3) via a MIT Reset pulse. The MIT Reset pulse is used during normal operation to clear out preprocessed replies stored in the secondary buffer if an expected ATCRBS/Mode S All-Call interrogation was not detected.

6.4.1.3.2 Memory Write Control Logic.

The memory write control logic, shown in figure 6.4.1.3.2-1, generates two identical sets of control signals; one set controlling the write sequence for buffer No. 1 and the other controlling the write sequence for buffer No. 2. The control line PRIMARY/SECONDARY SELECT determines which set of write control lines is selected. To which buffer the data is directed is under program control via a command instruction. The primary/secondary designation of the buffers is controlled by the 56 EARLY pulse and the designation is reversed each time the pulse is received from the UIT. The control line PRIMARY/SECONDARY SELECT is defined by these two conditions. For example, if buffer No. 1 and buffer No. 2 are respectively designated as secondary and primary, and data is directed to the primary buffer, the write control lines for buffer No. 2 will be selected. Since the timing sequence for both buffers is identical, only the primary buffer write sequence will be addressed in the following discussions.

The write control sequence requires four control signals to properly load data into the memory buffer. A typical write cycle to the primary buffer is shown in figure 6.4.1.3.2-2. When the computer executes a Data Available instruction, the DAG0 control pulse is generated. First, this pulse is synchronized to the 4-MHz system clock through D flip-flops IC_1 and IC_2 before the write control signals are generated. The synchronized DAG0 signal is fed to pulse shaping and delay logic to generate three of the write control signals; MEMORY ADDRESS MULTIPLEXER SELECT, WRITE BUFFER ENABLE, and MEMORY BUFFER WRITE ENABLE. The pulse shaping and delay logic consists of a 100-ns digital delay line and logic gates A through E. The memory write address is forwarded to the memory address bus when the MEMORY ADDRESS MULTIPLEXER-1 SELECT control line goes high. The data on the I/O bus are forwarded to the memory I/O data bus when the WRITE BUFFER-1 ENABLE control line goes active low, 20 ns later. The MEMORY BUFFER-1 WRITE ENABLE control signal goes active low for 40 ns, 10 ns later, loading the data into the memory location specified by the memory write address. On the lead edge of the following clock cycle, the write address counter is incremented to the next address via control signal WRITE ADDRESS COUNTER ENABLE which is generated by the lead edge detection logic consisting of D flip-flop IC_2 and AND gate F. Control signal WRITE ADDRESS COUNTER-1 ENABLE remains active high for one clock cycle, each DAG0 signal received.

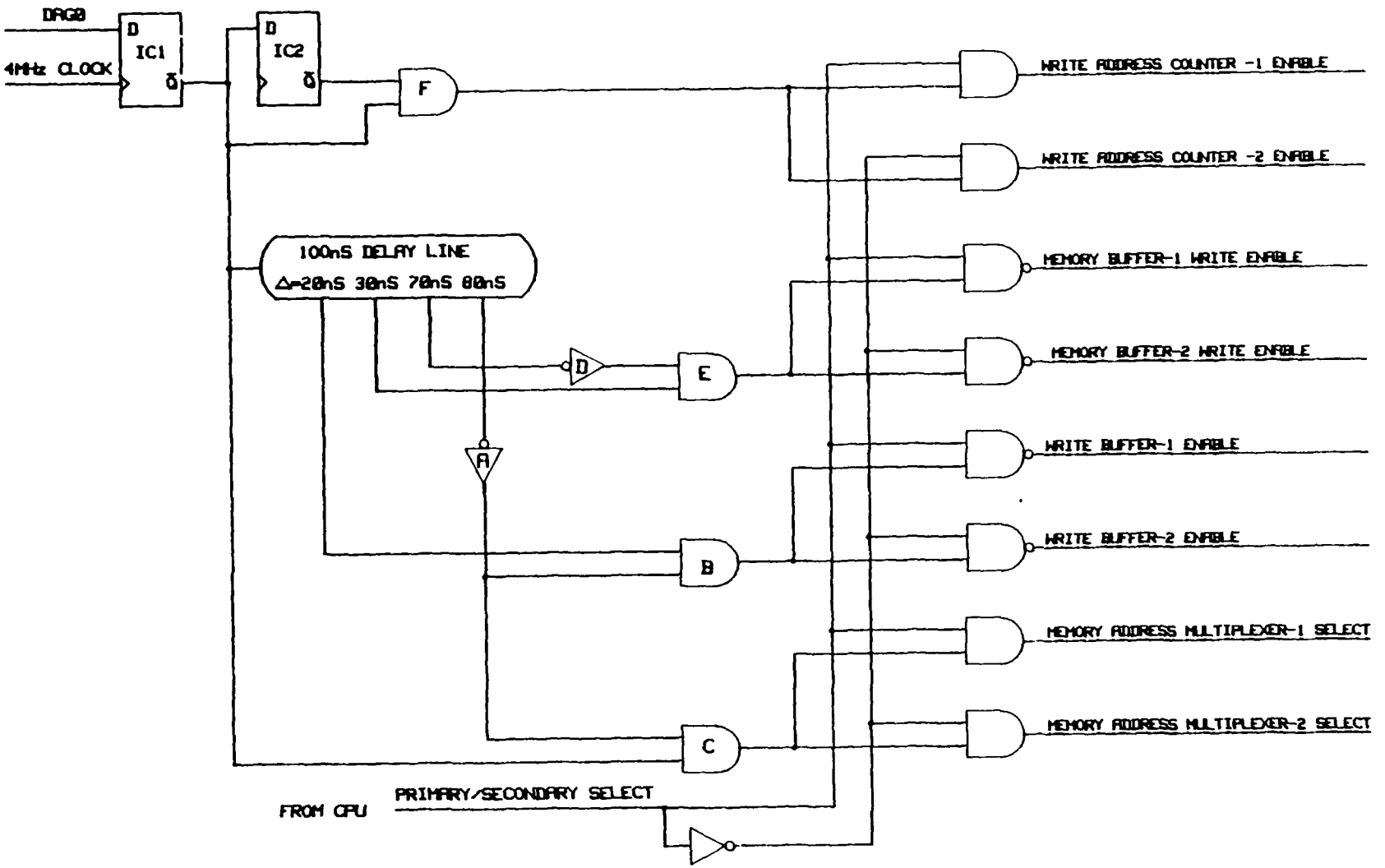


FIGURE 6.4.1.3.2-1. MEMORY WRITE CONTROL LOGIC

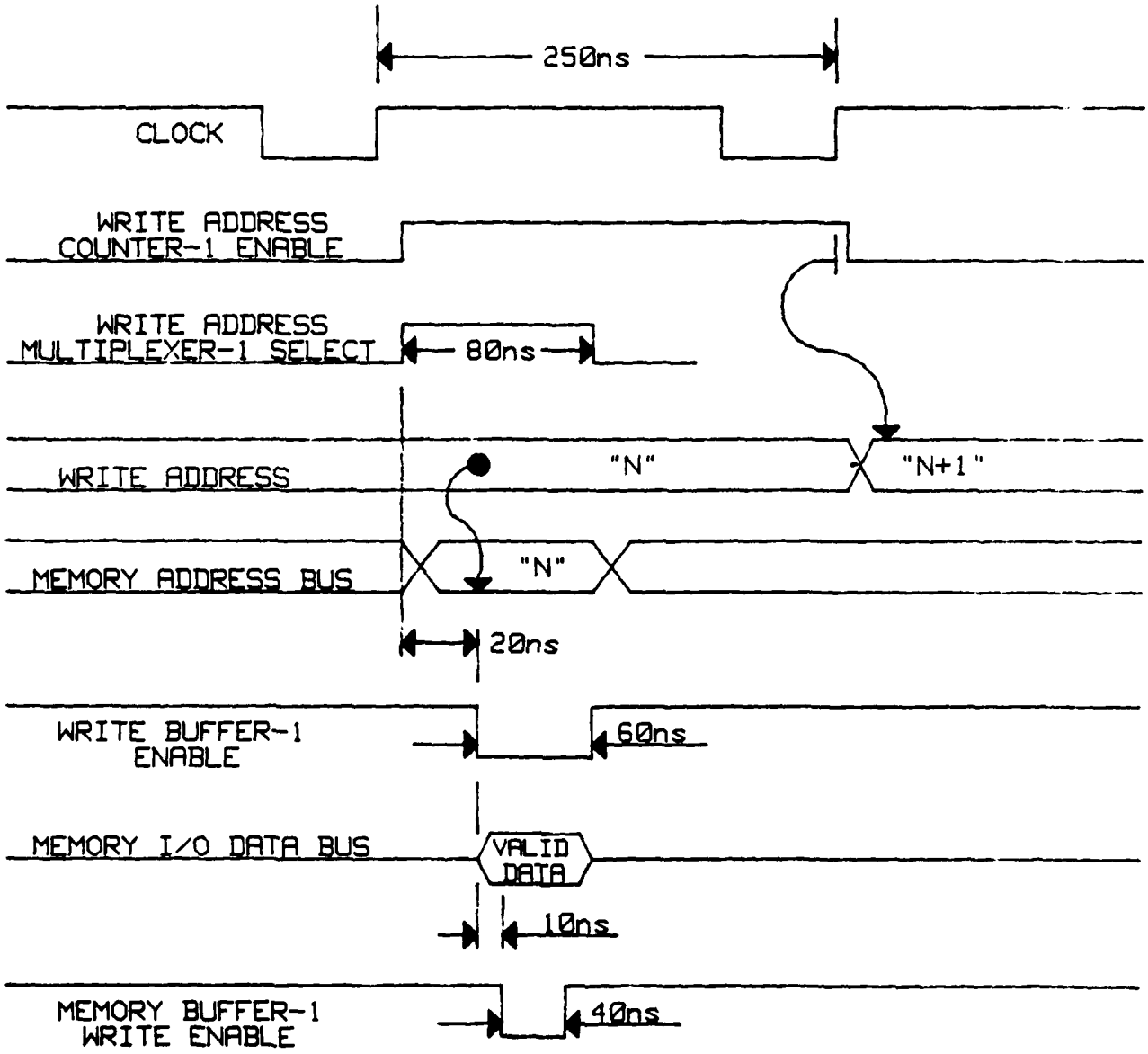


FIGURE 6.4.1.3.2-2. MEMORY WRITE CYCLE WAVEFORM

The primary memory can be accessed by the computer and controller on the same 4-MHz clock cycle without interference. The controller reads data from the memory during the positive transition of the clock signal (low-to-high). Therefore, the period immediately after this transition can be used for writing if the memory access time is small and the read and write address can be set up fast enough, that is, within one clock cycle. This is accomplished by high speed memory and Schottky support logic used. The Am2148-35 memory chip used in this design has a worst case access time of 35 ns providing fast enough speed for reading and writing in a 250-ns clock cycle.

6.4.1.3.3 Memory Read Control Logic.

The controller input bus (D-Bus) is tied to two sets of open-collector drivers shown in figure 6.4.1.3-1. The input of one set is connected to the data I/O bus of memory No. 1 and the other to the data I/O bus of memory No. 2. In any particular instance, only the set of drivers connected to the primary memory will be allowed to output data to the D-Bus. Similarly, the computers input bus (DIN-Bus) is tied to two sets of tristate drivers. The input of one set is connected to the data I/O bus of memory No. 1 and the other to the data I/O bus of memory No. 2. Either set of drivers will be allowed to output data to the DIN-Bus, independent of the primary/secondary assignment. All of the driver output enable lines and the increment read address control lines are generated by the memory read control logic shown in figure 6.4.1.3.3-1.

During the normal mode of operation, the computer is prohibited from reading both memory buffers and the MRC is restricted to reading from the primary memory. This is accomplished as follows. When control signal I/F DIAG1 is low, the input to multiplexer M_1 is taken from two output lines of multiplexer M_2. Only one of the output lines is active at any one time and is selected by control signal ASSIGN PRIMARY. When control signal ASSIGN PRIMARY is low, memory buffer No. 1 is primary, or when it is high, memory buffer No. 2 is primary.

The MRC reads data from the primary memory by sending a READ4 (READ6) command to the MRG Interface Buffer. When the READ4 (READ6) command is sent, counter A is preset to 12 (10) and increments for 4 (6) clock cycles until its count raps back to 0. During this period, counter output line Q_D is high placing the output of the primary buffer on the D-Bus and incrementing the primary read address counter by 4 (6).

However, it is highly desirable for diagnostic purposes to check the contents of the memory after known data has been written into it. Therefore, the logic is required to allow the computer to read the contents of either memory, independent of their primary/secondary assignments.

When the MRG Buffer Interface is placed under diagnostic operation and signal I/F DIAG1 goes active high, control of the read address counters is switched over to program control. Also, AND gates A and B are disabled, so that the bus drivers that forward data to the Controller are inhibited. Now data can be accessed from either memory by the computer. The PRIMARY/SECONDARY SELECT control line determines which buffer is read: if set low, the designated primary buffer is selected; if set high, the designated secondary buffer is selected. The data is read when the computer sends the data request to the MRG. In order to read the next location, the INCREMENT READ ADDRESS COUNTER pulse must be generated. This pulse is passed to the designated primary read address counter, incrementing it by 1.

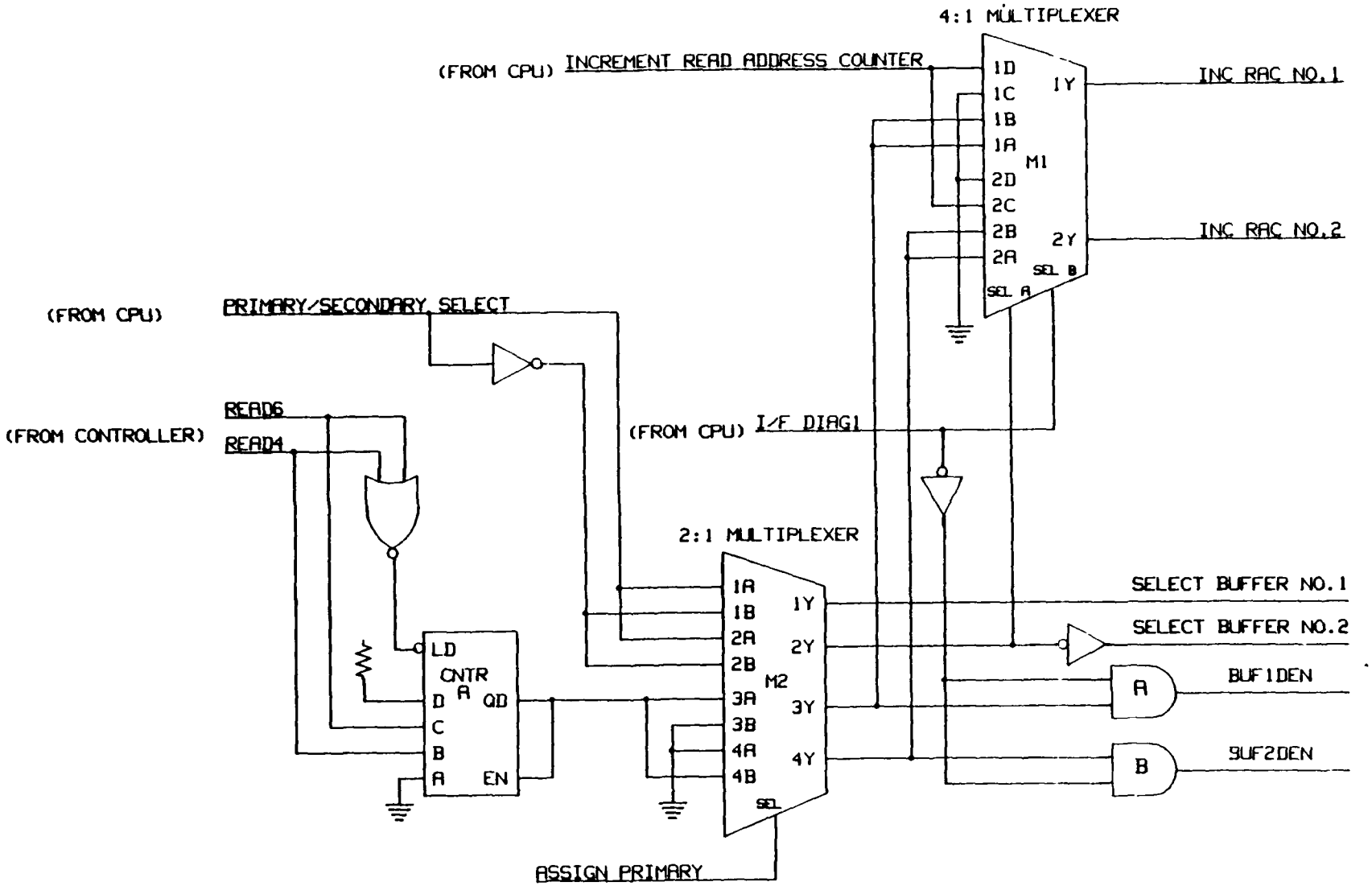


FIGURE 6.4.1.3.3-1. MEMORY READ CONTROL LOGIC

6.4.1.3.4 Reply Counters and Support Logic.

Two reply counters are provided as shown in figure 6.4.1.3.4-1 to keep track of the number of replies stored in the buffer memories. These counters are 8-bit up-down type, and at any instant one is associated with the primary memory and the other with the secondary memory. Hence, they are referred to as the primary reply counter and the secondary reply counter. The output of each reply counter is connected to the P-input of an 8-bit magnitude comparator, with the Q-input tied low. The P=Q function of the comparators are connected to the input of a 2:1 multiplexer which is used to forward the status of the primary reply counter to the controller. When the reply count is 0, meaning no replies stored in the memory, the P=Q function will be high.

Both reply counters can be read by the computer under diagnostic operation. A 16-bit wide 2:1 multiplexer is used for this purpose. The output of reply counter No. 1 is connected to the A-input of the lower byte of the multiplexer and to the B-input of the upper byte. Similarly, the output of reply counter No. 2 is connected to the B-input of the lower byte of the multiplexer and the A-input of the upper byte. Using this configuration, the output of the multiplexer will always represent the same fields, that is, the lower byte will always represent the primary reply count and the upper byte will always represent the secondary reply count, independent of the primary/secondary designations given to reply counter No. 1 and reply counter No. 2.

The computer can increment these counters at any time and when in the diagnostic mode has the capability of decrementing them as well. However, in the normal mode of operation, the controller can decrement only the reply counter designated as primary at the time.

Figure 6.4.1.3.4-2 shows the control logic that performs these operations. Assuming memory buffer No. 1 is selected as the primary buffer (i.e., ASSIGN PRIMARY BUFFER = 0) the A-side of the 2:1 multiplexer will be forwarded to the reply counters. If the PRIMARY/SECONDARY SELECT line is high, the INCREMENT REPLY COUNTER pulse is fed forward to the 3A input of the multiplexer generating a high pulse, for one clock cycle on the UP/DWN RC-1 line, which controls the direction of the count. This pulse is connected to the up/down control pin of reply counter No. 1 and is fed forward to the count enable pin of reply generator No. 1 through an EXCLUSIVE OR gate to generating the active low pulse ENABLE RC-1. Under the same circumstances, if the pulse DECREMENT REPLY COUNTER is generated by either the controller or the computer, it is forwarded through the 1A input of the 2:1 multiplexer through the same EXCLUSIVE OR gate generating the active low pulse ENABLE RC-1. At this instant, the UP/DWN RC-1 line is low, causing the counter to decrement. If the pulse DECREMENT REPLY COUNTER generated by the controller and the pulse INCREMENT REPLY COUNTER generated by the computer, occur simultaneously, the EXCLUSIVE OR gate will prevent reply counter No. 1 from counting in either direction.

In a similar fashion, if the PRIMARY/SECONDARY SELECT line is low, the INCREMENT REPLY COUNTER pulse is fed to the 4A input of the multiplexer to increment reply counter No. 2 designated as the secondary reply counter. Note that the DECREMENT REPLY COUNTER generated by the controller is not forwarded to the 2A input of the multiplexer, therefore, the secondary reply counter cannot be decremented by the controller. However, under diagnostic operation, the computer is allowed to decrement the secondary reply counter.

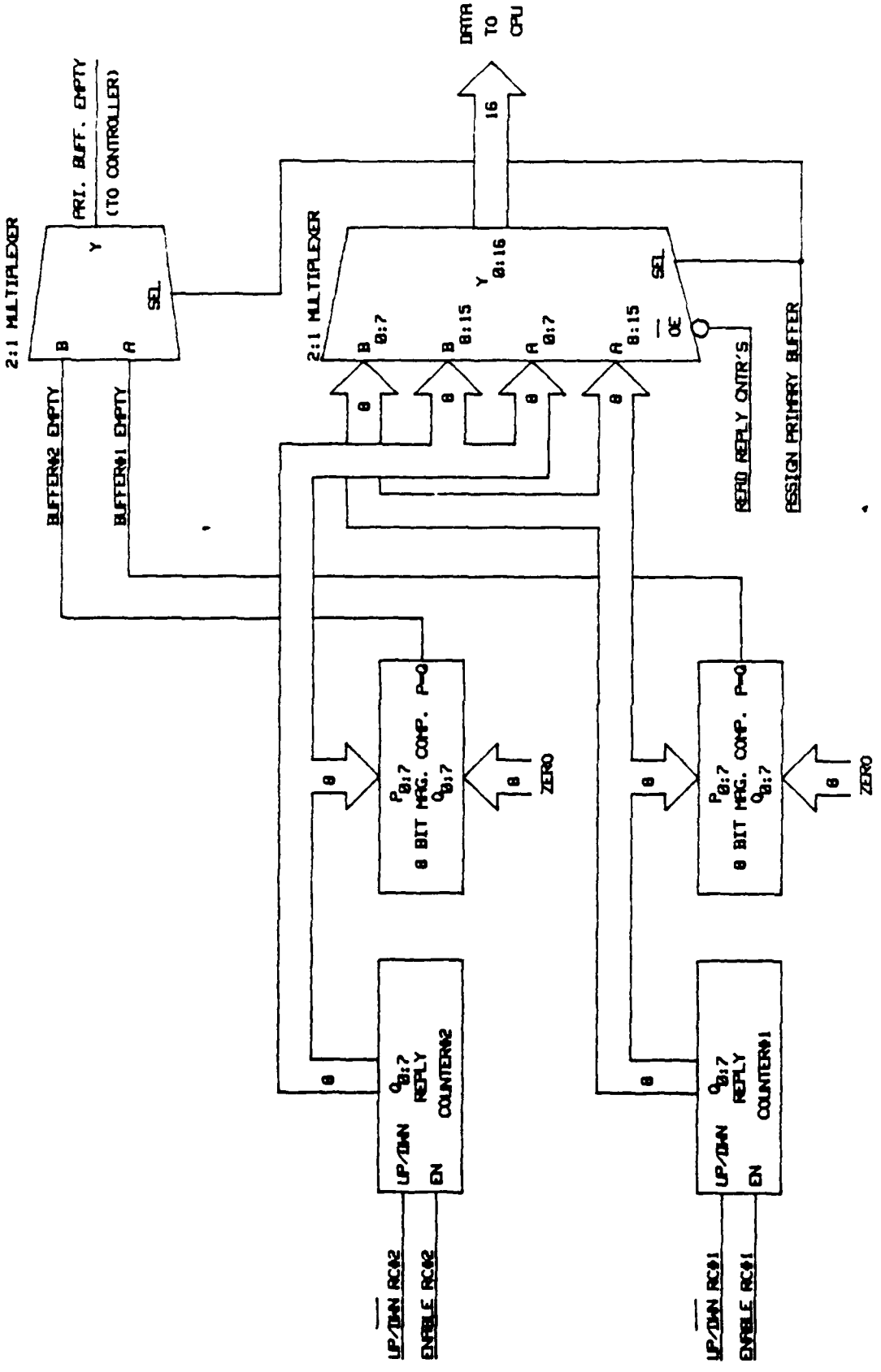


FIGURE 6.4.1.3.4-1. REPLY COUNTERS AND SUPPORTING LOGIC DIAGRAM

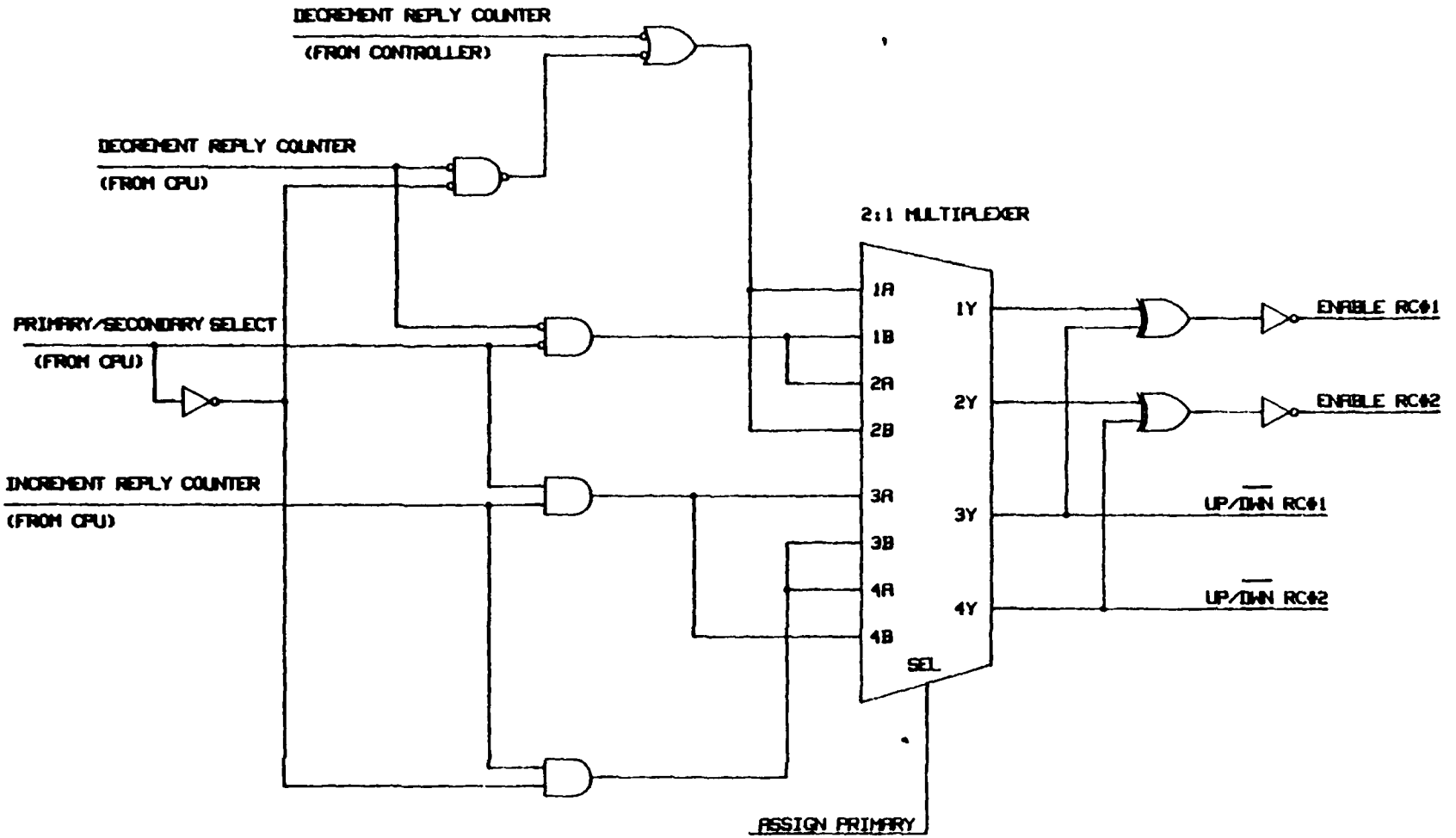


FIGURE 6.4.1.3.4-2. REPLY COUNTER CONTROL LOGIC

Each reply counter can be reset to 0 by any one of the following conditions: (1) during system power up, (2) under program control via reset command, and (3) via a MIT Reset pulse. (For more detail, see section 6.6, Missing Interrogation Timer.)

6.4.1.3.5 Operating Mode Decode Logic.

The Operating Mode Decode logic provides the means of selecting the operating mode of the MRG and controlling the various functions under each mode of operation. This logic consists of a doubled buffered register set and four single-shot pulse generators as shown in figure 6.4.1.3.5-1.

The doubled buffered register set is used to latch 8 bits of the asynchronous command (register A) sent by the computer, and synchronize the command bits (register B) to the 4-MHz system clock used by the MRG logic. The output of register B includes the 3-bit operation field which is forwarded to the 3-input decoder C that sets up the MRG Buffer Interface to the correct support configuration.

The four single-shot pulse generators produce the control pulses; RESET, INCREMENT-REPLY-COUNTER, INCREMENT-READ-ADDRESS-COUNTER, and DECREMENT-REPLY-COUNTER. A single pulse is generated when the appropriated command bit is set, every time a command is received. After the pulse is generated, the single-shot pulse generator is reset ready for the next command. The timing waveform of this generator is illustrated in figure 6.4.1.3.5-2. This same circuit design is used repeatedly in other device control circuitry. Therefore, it is appropriate to discuss it in detail at this time.

This circuit works as follows. (Refer to figure 6.4.1.3.5-1.) Assuming that the desired command bit is set, it is latched in D flip-flop IC_1 on the low-to-high transition of control line CMG0. The Q-output of flip-flop IC_1 is clocked into D flip-flop IC_2 to synchronize it to the MRG logic (4-MHz system clock). At this time, the output of AND gate A goes high since both of its inputs are high. On the following clock cycle, the output of AND gate A is loaded into D flip-flop IC_4 placing a low on its /Q-output. This line is returned to the asychronous clear of flip-flop IC_1 resetting it for the next command. Simultaneously, the Q-output of flip-flop IC_2 is loaded into D flip-flop IC_3, forcing its /Q-output low which in turn forces the output of gate A low generating the control pulse.

Six command bits are forwarded to the MRC as part of the MRG Interface status word. Included in these bits is the operation control field which the MRC uses to determine whether it is in the normal operating mode or in one of the diagnostic support modes as defined in table 6.4.1.3.5-1.

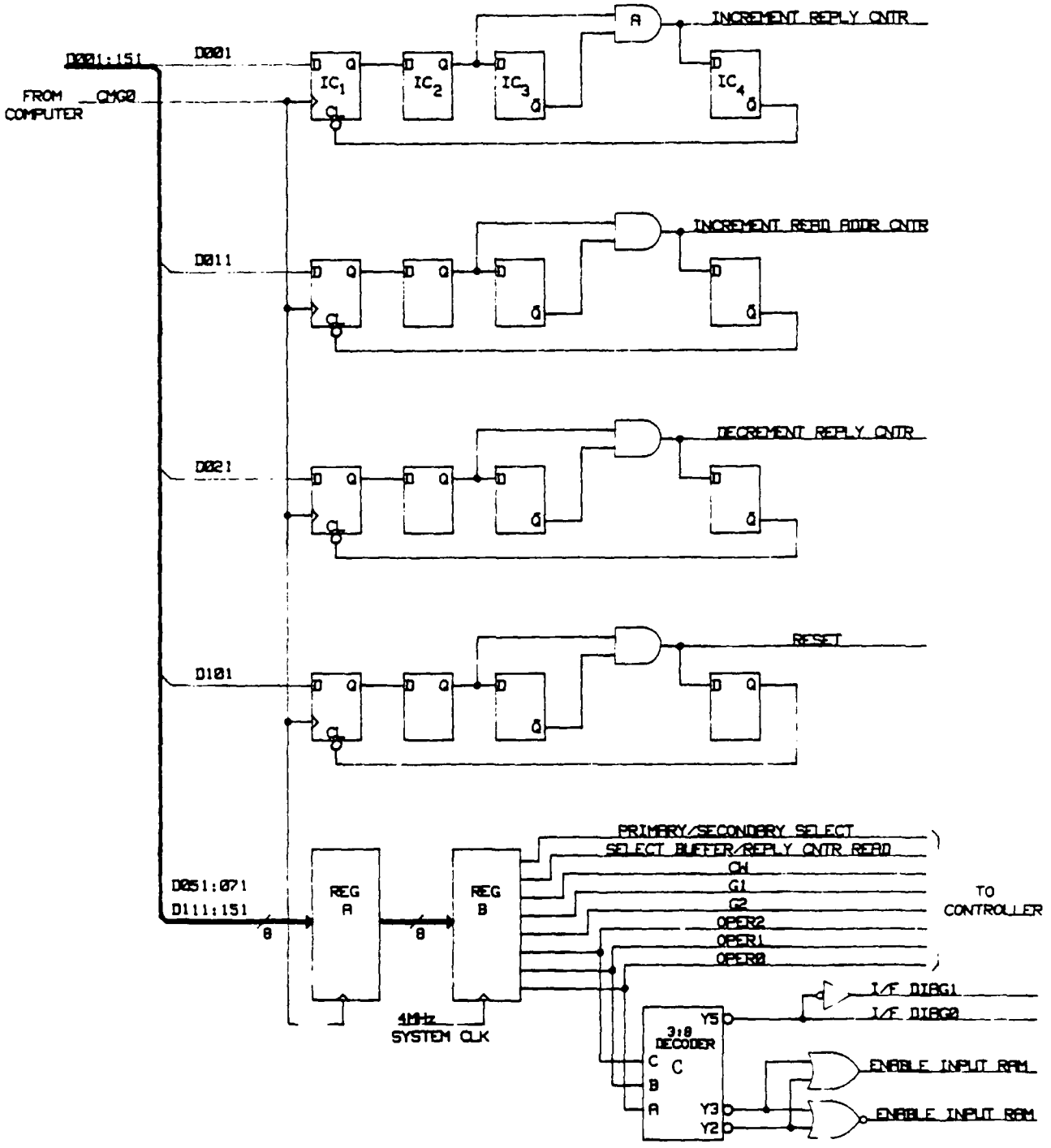


FIGURE 6.4.1.3.5-1. MRG COMMAND DECODE LOGIC

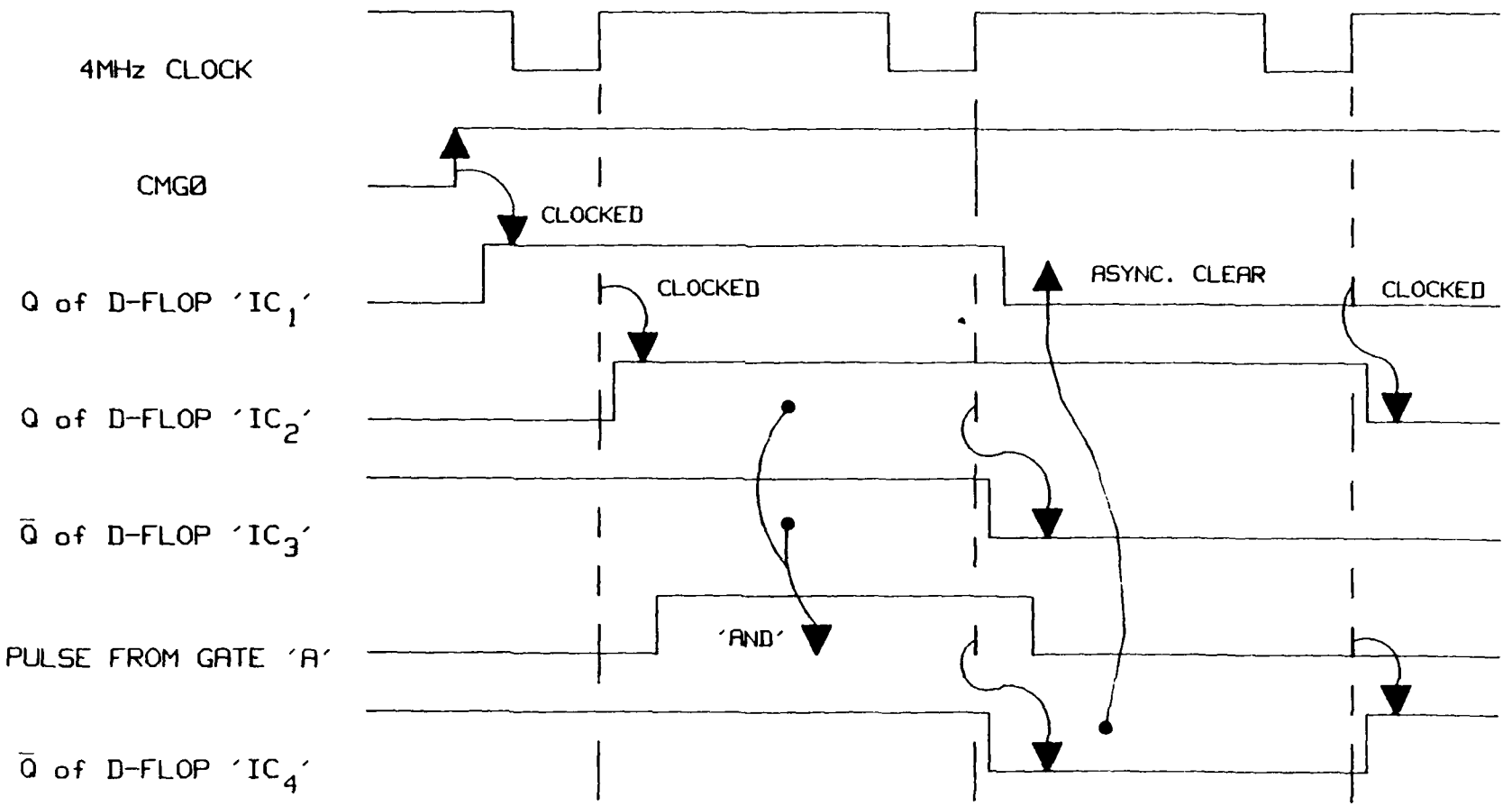


FIGURE 6.4.1.3.5-2. SELF RESETTING PULSE GENERATOR TIMING WAVEFORM

TABLE 6.4.1.3.5-1. MRG MODE-OF-OPERATION CONTROL FIELD

OPERATION FIELD BITS			MODE OF OPERATION
13	14	15	
0	0	0	Normal
0	0	1	RF Diagnostic Loop-Test
0	1	0	MAT Diagnostic
0	1	1	Microprocessor Diag.
1	0	1	Buffer Interface Diag.

As in the operational mode, each diagnostic mode requires a special microcode routine. These routines are covered in Volume II, appendix B.

6.4.1.3.5.1 Normal Mode.

When the normal mode is selected, the command decode logic inhibits all of the diagnostic control signals and allows the computer to only load reply data into the primary and secondary buffers and increment their associated reply counters. The MRC can read reply data from the primary buffer and decrement the primary reply counter only. The MRC then routes the reply blocks to the MATs for transmission.

6.4.1.3.5.2 Buffer Interface Diagnostic Mode.

This diagnostic mode provides the capability of checking the Buffer Interface circuitry independent of the rest of the MRG logic. This is accomplished when the 3-input decoder C (refer to figure 6.4.1.3.5-1) decodes a 5 in the operation control field and produces the interface control signal I/F DIAG. This signal, as discussed in section 6.3.1.3.3, switches full control of both primary and secondary buffers to the computer and inhibits the MRC from accessing the primary buffer when data is available. In this mode all four control pulses, generated by the single-shot pulse generators and control lines PRIMARY/SECONDARY SELECT and SELECT BUFFER/REPLY-CNTR READ, are now functional allowing the computer to load (read) both memory buffers independent of primary/secondary status, and increment (decrement) both reply counters and read the results.

6.4.1.3.5.3 Microprocessor Diagnostic Mode.

The Microprocessor Diagnostic mode provides a way of checking the Microprocessor hardware and the buses between the MRC and the Buffer Interface circuitry. In this mode, the interface circuitry configures itself to accept reply data from the computer as if in the normal mode. Likewise, the MRC fetches the data blocks as if in the normal mode but does not transfer the data to the MATs but back to the computer. However, this requires a means of transferring data from the MRC to the computer to check out the MRC. (This circuitry is also used under the MAT Diagnostic mode to be discussed in section 6.4.1.3.5.4.) This is accomplished by the Input Data Circuitry shown in figure 6.4.1.3.5.3-1. The Command Decode Logic controls the Input Data Circuitry.

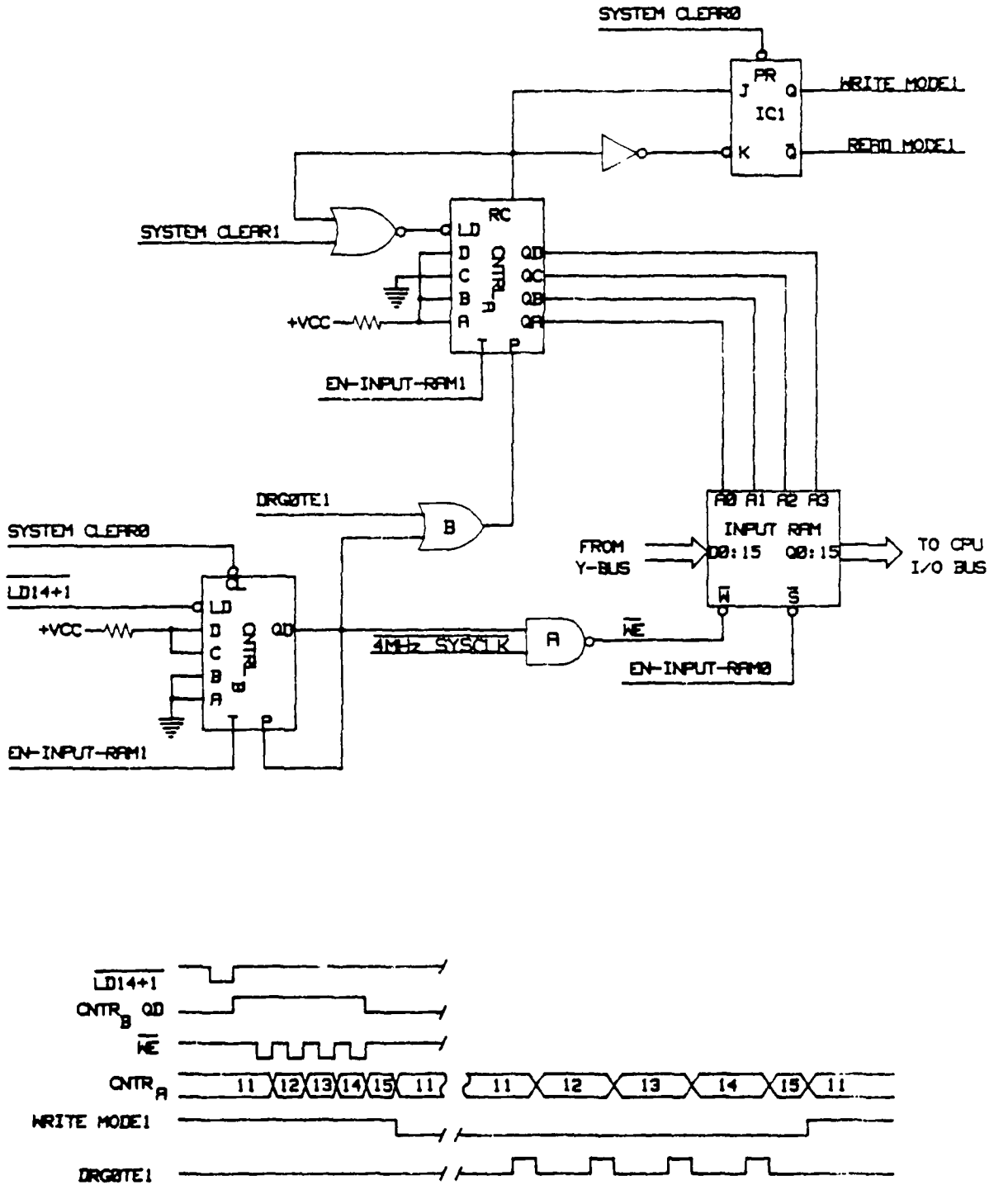


FIGURE 6.4.1.3.5.3-1. INPUT DATA CIRCUITRY AND TIMING WAVEFORMS

The Input Data Circuitry, as shown in figure 6.4.1.3.5.3-1, consists of a 16-bit x 16-word RAM, a RAM address counter, a read/write mode flip-flop, and a load control counter. Note that only the locations 11 through 14 of the RAM are used, so only four-word data block transfers are handled. The input of the RAM is tied to the Y-Bus of the MRC, and the output of the RAM is fed to the return bus (DIN-001:151) of the computer. The operation of this circuitry is given below.

Assume that the Input Data Circuitry has just been initialized and the MRG placed in the Microprocessor Diagnostic mode. The RAM address counter A is loaded with a value of 11 and the load control counter B is reset to 0. Both counters are conditionally enabled by interface control signal ENABLE-INPUT-RAM1, the RAM's output is placed on the DIN-Bus by interface control signal ENABLE-INPUT-RAM0, and the read/write mode flip-flop IC_1 is set to its write mode. Both the computer and the MRC monitor this flip-flop to determine the RAM's availability.

When data is to be loaded into the RAM, the MRC first checks to see whether the Input Control Circuitry is in the write mode and then sends a load command (LD14) on the C-Bus followed by four data words on the Y-Bus. A 12 count is loaded into counter B and its Q_D output goes high. This signal is fed back to the P-Enable input of counter B allowing it to increment until it overflows at which time the Q_D output goes low placing counter B back at 0. Therefore, Q_D is high for four clock cycles. During this period, NAND gate A is enabled producing four write enable (WE) pulses to the RAM and the Q_D signal is digitally coupled to the P-Enable input of counter A through OR gate B incrementing the counter after eacn write to the RAM. After the fourth word is loaded, the carry output of counter A goes high causing, on the next clock cycle, the read/write flip-flop to toggle to the read mode and counter A to be loaded back to 11.

During this time, the computer monitors the read/write flip-flop until it toggles to its read mode indicating that data is available. Note that there is no protection logic that would prevent the computer from attempting to read the RAM when the Input Data Circuitry is in the write mode except for properly monitoring the state of the read/write flip-flop. The computer then issues four data requests. Counter A is stepped to the next address after each word is read. After the fourth read takes place, the carry pulse is generated causing, on the next clock cycle, the read/write flip-flop to toggle back to the write mode and counter A to be loaded back to 11. This sequence is repeated until all data are transferred from the MRC to the computer. The waveform diagram shown in figure 6.4.1.3.5.3-1 illustrates the write/read cycles of the Input Control circuitry.

6.4.1.3.5.4 MAT Diagnostic Mode.

In the MAT Diagnostic mode, the Interface Buffer is placed in the same support configuration as in the Microprocessor Diagnostic mode. The Input Data Circuitry is used the same way to return data from the MRC to the computer. The main difference in this mode is the routing of the reply block once the MRC reads it from the primary memory buffer. Instead of returning it immediately to the Input Data Circuitry, the MRC loads the reply into a selected MAT, as directed by the 2-bit field, G1 and G2, shown in table 6.4.1.3.5.4-1. Note all four reply generators are checked even though the fourth generator is not used in the MRG. This is done so that the complete reply generator board is checked. This is necessary since the reply generator boards are identical and are interchangeable.

TABLE 6.4.1.3.5.4-1. MAT SELECTION CONTROL FIELD

REPLY DIRECTION FIELD		DESTINATION
11	12	
0	0	MAT No. 1
0	1	MAT No. 2
1	0	MAT No. 3
1	1	MAT No. 4

Special diagnostic hardware built into each reply generator allow the MRC to read back the reply control fields (reply time, power, monopulse L/R bit, M/S bit, and type fields) loaded into the Hold 2 register at the time the code train was transmitted. Also a serial-to-parallel converter is used to collect the PAM modulation code train pulses during transmission.

The Read Control Logic in the reply generator that supports this operation, along with its timing waveform, is shown in figure 6.4.1.3.5.4-1. The MRC sends a read command (READ1-4) to the desired reply generator, then reads the data on the D-Bus. The READ1-4 command is lead-edge detected by the reply generator then inserted into a 4-bit shift register. As the pulse is shifted through the register, the control signals ENABLE-WORD 1, 2, 3, and 4 are produced. These control signals are used to reassemble the reply data in the same format as it was received placing one word (16-bits) at a time on the D-Bus to be returned to the MRC. Once the data block is received from the MAT, the MRC then transfers it to the Input Data Circuitry in the identical manner as described in section 6.4.1.3.5.3.

6.4.1.3.5.5 RF Diagnostic Mode.

Lastly, the RF Diagnostic mode provides a means of checking out the analog circuitry of the MRG along with some of the MAT digital circuitry not checked under the MAT diagnostic mode. This is possible by using the STU to observe and sample the generated replies. The RF Diagnostic mode is similar to that of the MAT Diagnostic mode except no data is returned to the ARIES computer; therefore, the Input Data Circuitry is not needed.

6.4.2 Fruit Reply Generator.

The FRG consists of the FRC, four target reply generators (referred to as the FAT generators), the ARPG, the MRPG, and the FRG Interface. Three of the FATs are dedicated to generating ATCRBS fruit and the fourth FAT is dedicated to generating Mode S fruit. A block diagram of the major modules of the FRG is presented in figure 6.4.2-1. The FRG digital circuitry is mounted on five boards installed in the ARIES digital chassis. (See table 5-2 for board titles and slot numbers.)

6.4.2.1 Fruit Reply Controller.

The hardware for the FRC is identical to that of the MRC. However, the firmware resident in the ROM does differ. (Refer to Volume II, appendix B, FRG Microcode Program Listings.)

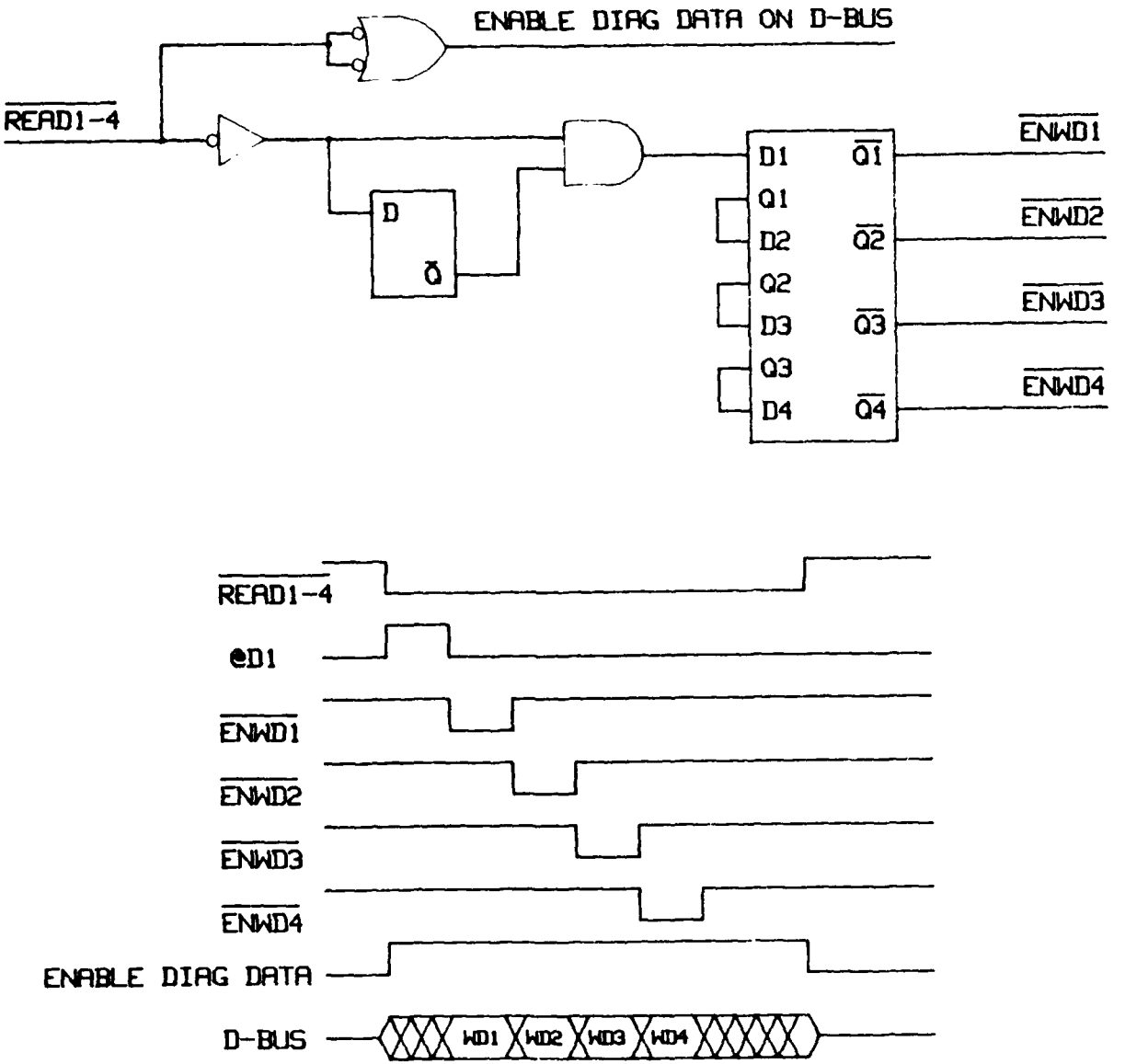


FIGURE 6.4.1.3.5.4-1. INPUT DATA CIRCUITRY READ CONTROL LOGIC AND TIMING WAVEFORMS

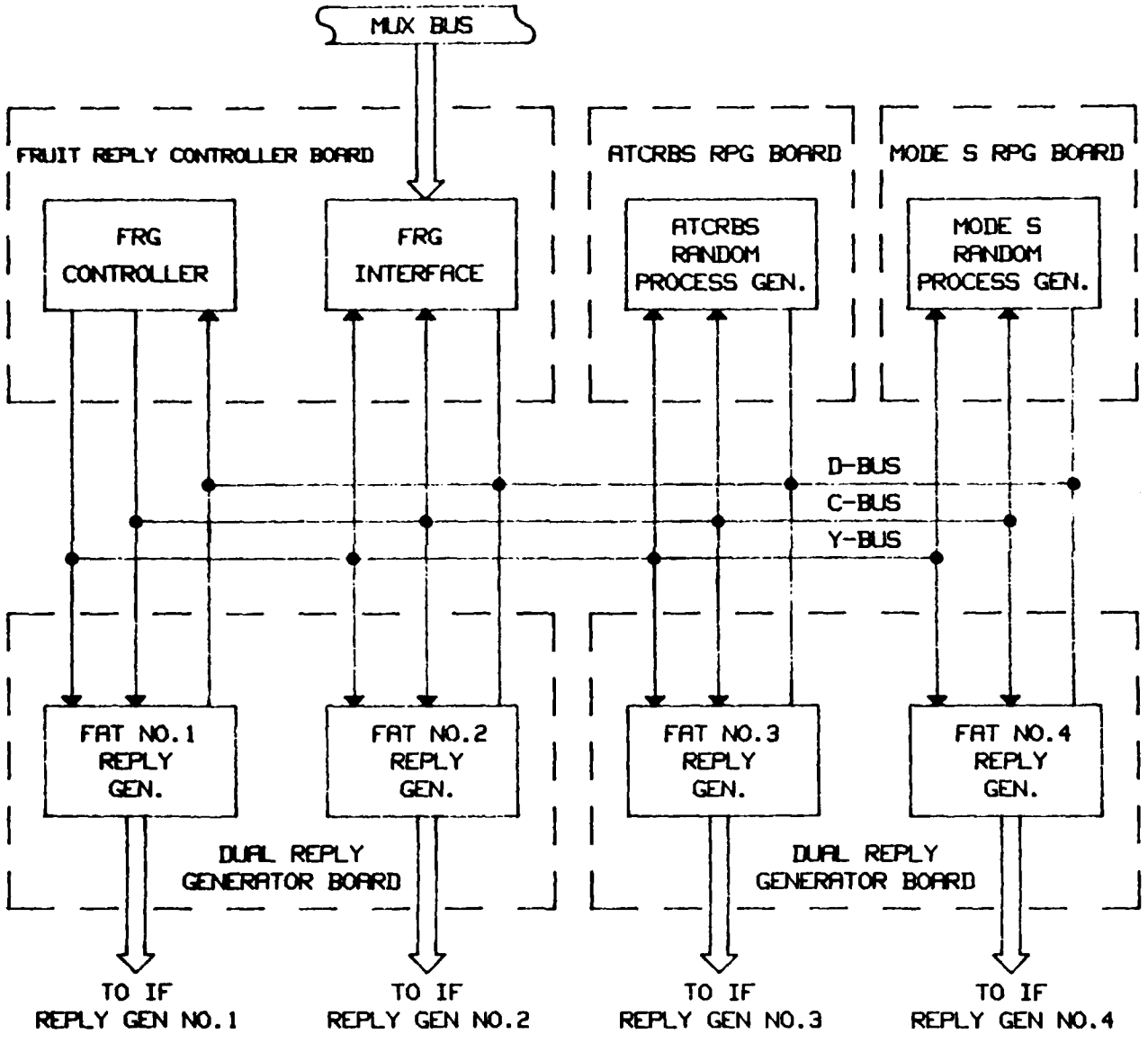


FIGURE 6.4.2-1. FRG DIGITAL CONFIGURATION

6.4.2.2 Fruit ARIES Targets.

The description of the FAT hardware is identical to the description of the MAT hardware so it will not be repeated. However, it is important to notice the differences in setting and clearing the GO-2 flip-flop for various cases. In the case of the MATs, the GO-2 flip-flop is set by the TOAR pulse from the Uplink Receiver and reset by the 56 EARLY pulse from the UIT as described previously.

In the case of the ATCRBS FATs, the GO-2 flip-flop is not set by the TOAR pulse but by the FTOAR compare pulse from the adjacent FAT. All three ATCBRS FATs are connected in daisy-chain fashion as shown in figure 6.4.2.2-1. The compare pulse from a FAT also clears its own GO-2 flip-flop. This scheme allows the FATs to fire in sequence. It is important that they do so as the FRC also sends replies to the FAT's sequentially. Therefore, the FRC can guarantee that the sum of any three successive reply times for the FATs will be greater than 20.3 μs duration of the ATCRBS reply with no SPI, thus no FAT will request that another reply be started until the current reply is complete.

In the case of the Mode S FAT, the GO-2 flip-flop is set all of the time. Also, the range clock rate for the Mode S FAT is different from the other FATs and MATs. The 4-MHz system clock is used instead of the 16-MHz system clock. The reduction in the range clock was necessary to achieve the low Mode S fruit rates specified. (The FRG is required to generate Mode S fruit between 10 to 500 fruit per second in 10-fruit per second increments.)

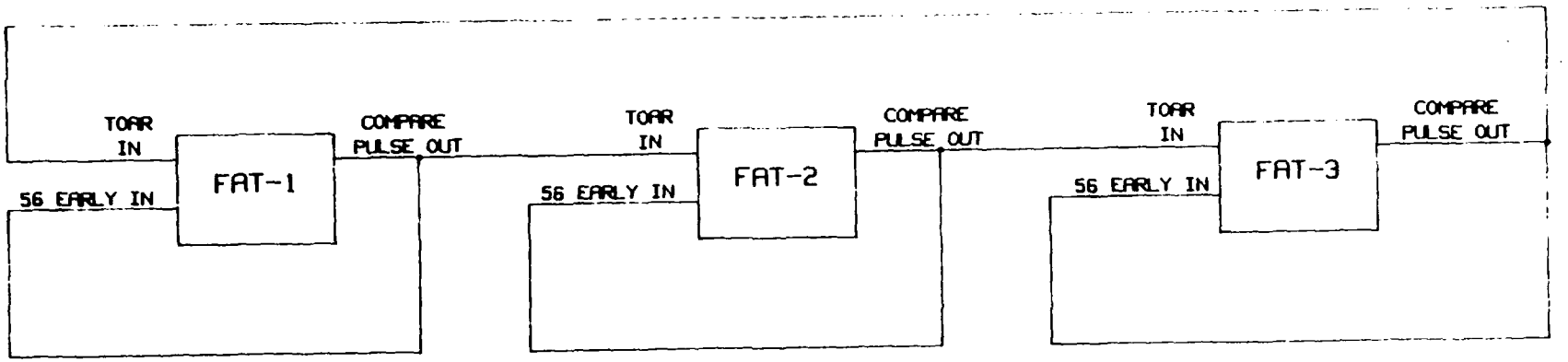
6.4.2.3 Random Process Generators.

The ARPG and the MRPG perform similar functions in generating fruit replies. Both construct groups of control words which determine the power, offboresight angle (including sign), mainbeam/sidelobe status, and data (ATCRBS code or Mode S data) of each fruit reply. The RPGs are also responsible for controlling fruit arrival times. To simulate real fruit each of these fruit parameters must vary randomly over its spectrum of possible values.

In the descriptions that follow, the details of operation for the ARPG precede the details of operation of the MRPG. However, the RPGs are functionally equivalent and in many cases the generation of the reply parameters are identical. The description of the identical modules are given once under section 6.4.2.3.1, the ATCRBS Random Process Generator, and differences only under section 6.4.2.3.2, the Mode S Random Process Generator.

6.4.2.3.1 ATCRBS Random Process Generator.

The ARPG constructs ATCRBS fruit reply blocks which are identical in format to those of the ATCRBS modeled reply blocks. These reply blocks are constructed with the uses of three fruit parameters which are used to control the operation of the ARPG: average fruit rate, fraction of fruit in the mainbeam (as opposed to being in the sidelobes), and fraction of fruit having a known, fixed, ATCRBS code (as opposed to having all code bits randomly selected). These three parameters are controlled by the operational software via the FRC.



NOTE: THE WIRING IS DONE ON THE BACKPLANE
OF THE DIGITAL CHASIS RATHER THAN
ON THE BOARDS THEMSELVES

FIGURE 6.4.2.2-1. DAISY-CHAIN OF THE ATCRBS FRUIT ARIES TARGET GENERATORS

As shown in figure 6.4.2.3.1-1, the ARPG consists of six control parameter generators, each responsible for generating one of the above FAT control word parameters. Also shown are the interface circuits in which the control word groups are assembled and through which they are transferred from the ARPG to the FRC. The data buses interconnecting the ARPG and the FRC are the C-Bus, the Y-Bus, and the D-Bus. Control commands transmitted via the C-Bus from the FRC to the ARPG are:

RESET: Resets all random number generators to a predefined value, and generates 16 new inter-arrival times.

READ: Causes a fruit control word group to be output to the FRC.

LOAD: Changes the average fruit rate, mainbeam/sidelobe ratio, and ratio of fixed code to random code.

Reading and loading are done over the D-Bus and the Y-Bus, respectively. Data formats for the fruit environment parameters are given in appendix D, under Fruit Reply Generator, of the ARIES Hardware Maintenance Manual.

6.4.2.3.1.1 Random Number Generators.

All random fruit parameters output by the ARPG are generated using pseudo-random number generators employing shift registers. Such devices called "linear maximal sequence code generators," consist of a basic shift register to which EXCLUSIVE-OR gates have been added. Outputs from the register stages are inputs to the EXCLUSIVE-OR gates, and the gate outputs fed back to other stages of the register to form single or multiple closed loops. When the register is clocked in a normal manner, the output of each stage forms a pseudo-random binary sequence.

The feedback connections of the EXCLUSIVE-OR gates determine the random sequence and whether the sequence is maximal or not. The maximal sequence generator cycles through all its 2^n-1 possible states, except "0," where "n" is the number of stages in the register. In the ARPG, all random number generators are connected to give the maximal sequence; also, only shift register stages numbered 7, 11, 15, and 17 are used. The 7-bit generator used in the ARPG is illustrated in figure 6.4.2.3.1.1-1.

Since 0 is an illegal state (causing a lockup condition), the generator must not start at this state. This requires a special form of reset circuit. Figure 6.4.2.3.1.1-2 shows this circuit. When a reset is decoded by the ARPG, counter A and counter B will be preset to 0 and 5, respectively. Note that one of the enable inputs of counter B is tied to the overflow of counter A; therefore, counter B is idle until counter A overflows. At that time, counter A is frozen at a maximum count (15) and counter B starts to increment from 5. When it reaches 8, it also freezes producing the waveform shown in figure 6.4.2.3.1.1-2. Note also that when the multiplexer signal select from the resetting circuit is high, the feedback path in the shift register is disconnected (see figure 6.4.2.3.1.1-1), since the multiplexer has selected the other input (load input from resetting circuit). With the feedback path disconnected, the random number generator operates as an ordinary shift register, allowing a known number to be loaded from the reset circuit. All random number generators are reset to "11100 - - - 00"; the number of 0's being equal to n-3, where n = number of stages. Resetting all

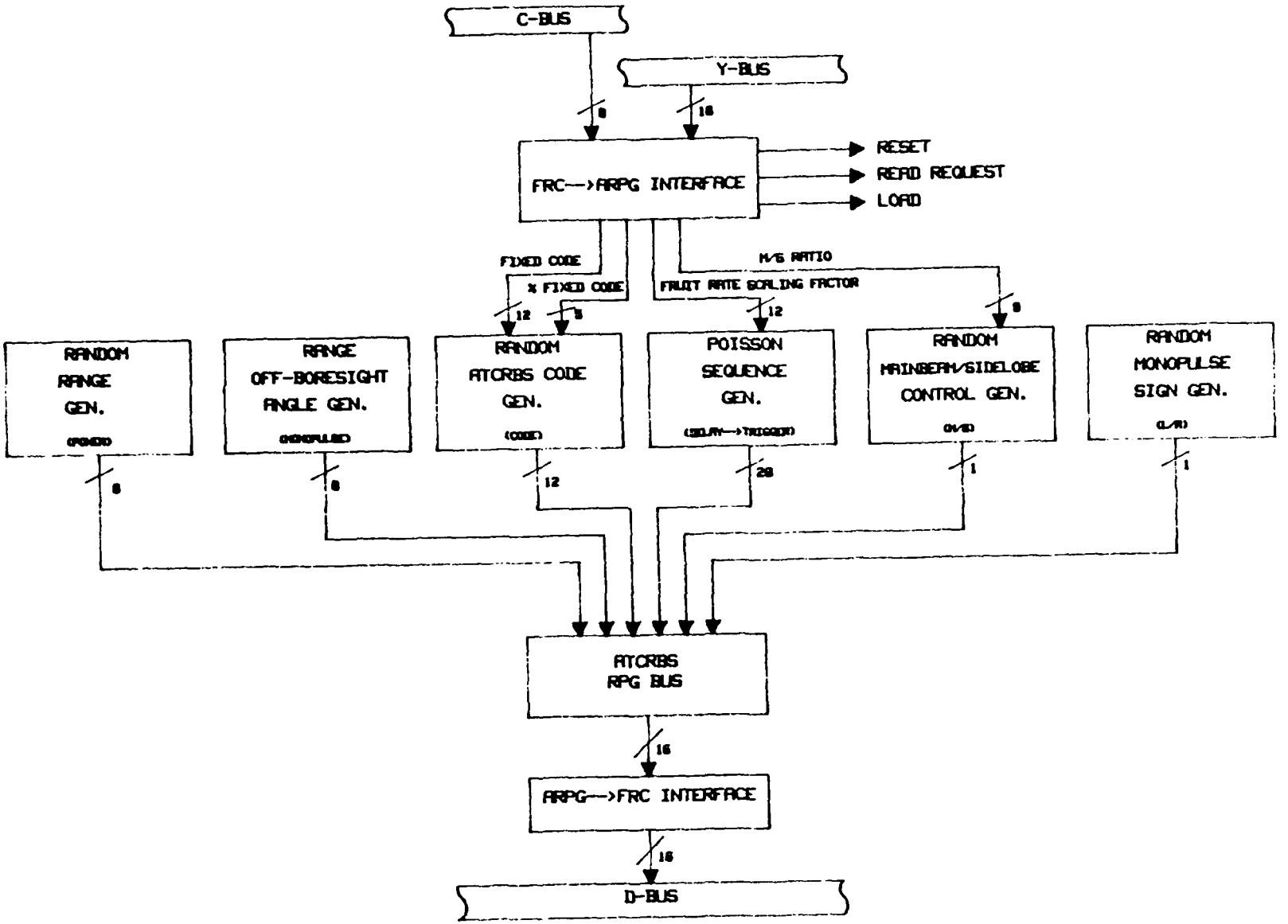


FIGURE 6.4.2.3.1-1. ARPG FUNCTIONAL BLOCK DIAGRAM

NO. BITS	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	TOGGLE INPUT TO N FLIP FLOP	RANDOM NUMBER GEN I.D. IN PRPG
7											*T	D	D	D	D	D	D	T_7=X1	2
11							T	T	D	D	D	D	D	D	D	D	D	T_{11}=X1	3,5
15			T	D	D	D	D	D	D	D	D	D	D	D	D	D	D	T_{15}=X1	1,4,6
17	T	T	T	T	T	D	D	D	D	D	D	D	D	D	D	D	D	T_{17}=X1	7

T=TOGGLE FLIP FLOP

D= DELAY FLIP FLOP

* EXCLUSIVE OR GATE IS ELIMINATED IN FEEDBACK PATH BECAUSE TOGGLE FLIP FLOP IMPLEMENTS AN EXCLUSIVE OR FUNCTION BETWEEN IT'S INPUT AND IT'S OUTPUT, $Q_{N+1}=Q_N \oplus T_N$

T_N=TOGGLE INPUT

Q_N=PRESENT STATE

Q_{N+1}= NEXT STATE

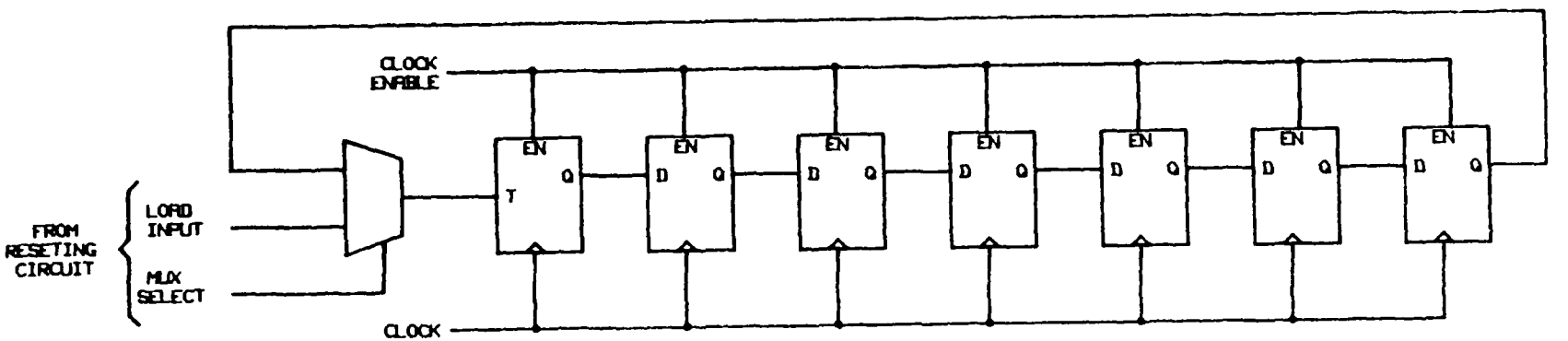


FIGURE 6.4.2.3.1.1-1. 7-BIT PSEUDO RANDOM NUMBER GENERATOR (RNG2)

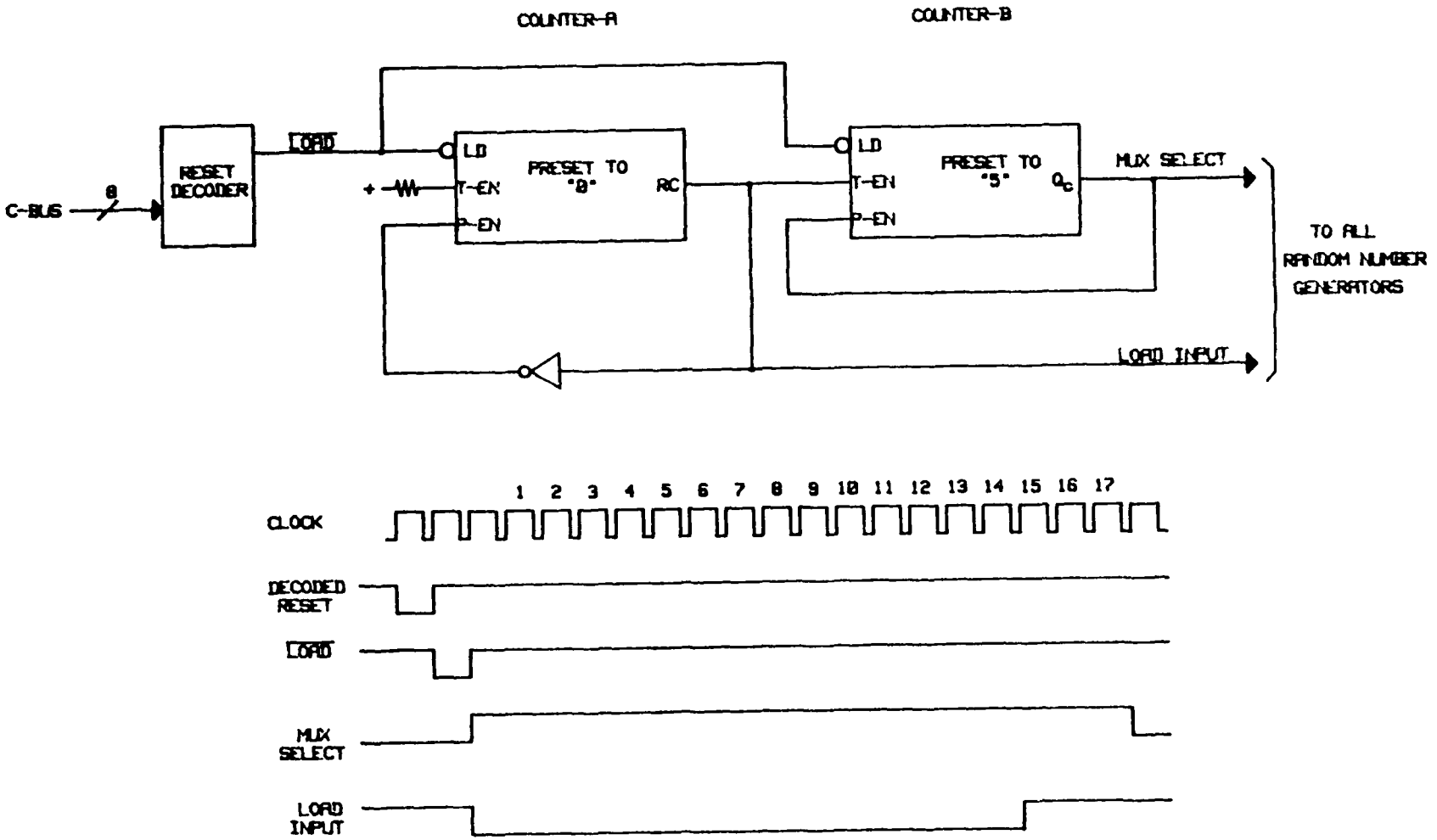


FIGURE 6.4.2.3.1.1-2. RNG RESETTING CIRCUITRY AND WAVEFORMS

generators to a known value at the beginning also allows ARPG diagnostics to be performed, as a computer simulation of the ARPG can also be started with the same set of conditions. ARPG data can be read by the computer using the ARPG diagnostic mode of the FRG Interface (see section 6.4.2.4).

6.4.2.3.1.2 Generation of Random Range (Power) Control Parameter.

The specification for the power level of ARIES generated fruit is as follows: "For mainbeam fruit, the SUM channel signal level (as referenced at the sensor's RF port) shall be randomly chosen according to the distribution;

$$P_{ml} = (-20 - 20*\log r_{ml}) \text{ dBm}$$

where r_{ml} is random and uniformly distributed between 1 and 100. For sidelobe fruit, the equivalent distribution is;

$$P_{sl} = (-55 - 20*\log r_{sl}) \text{ dBm}$$

where r_{sl} is random and uniformly distributed between 1 and 32. The resolution in power level shall be 1 dB.

This is equivalent to specifying a random mainbeam power level from -20 dBm to -60 dBm and a random sidelobe power level from -55 dBm to -85 dBm.

Two random number generators are used in the ARPG to meet these requirements. The Random Range Generator provides a 6-bit power field distributed according to either the mainbeam or sidelobe power specification. A single bit from the Random M/S Control Generator determines whether the reply is to be a mainbeam or sidelobe reply. The range (or power) field is used to control the main power attentuator of the analog portion of the FAT, giving mainbeam power levels of -20 to -60 dBm. If the M/S bit is set, an additional 35 dB of attentuation is switched to produce the sidelobe output distribution. Since a uniform distribution in range does not give a uniform power distribution, it is necessary to use a separate 15-bit random number generator, RNG4 (see figure 6.4.2.3.1.2-1) of which 8 bits are used to address a 512-word "Power ROM."

The 9th address bit for the ROM is the M/S bit, and this is used to choose between two distributions. This ROM modifies its uniform input distribution to provide the desired 6-bit power control field. The first 256 locations contain the mainbeam distribution, and the upper 256 locations contain the sidelobe power distribution. The remaining discussion addresses the question of what the ROM content should be.

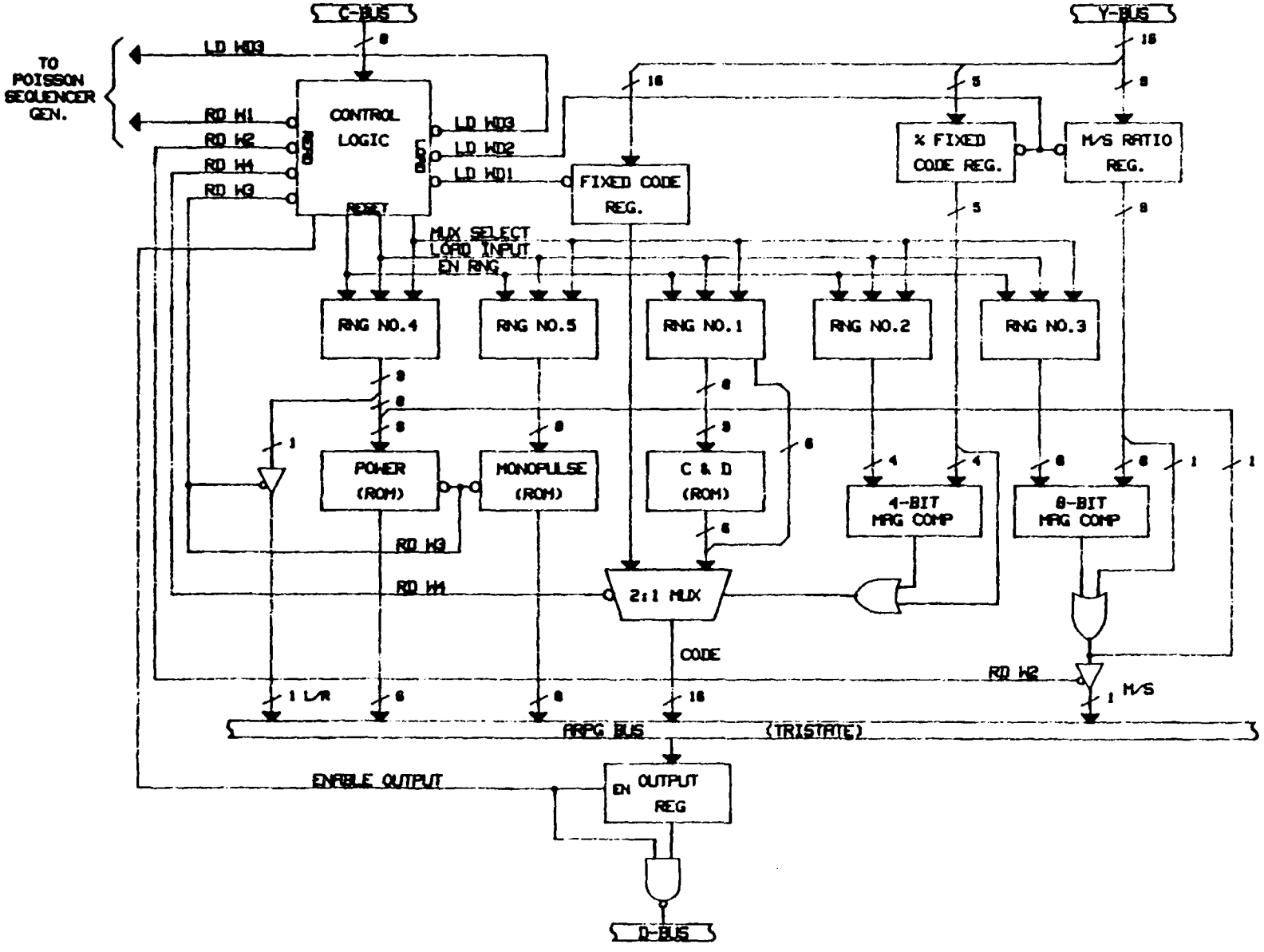


FIGURE 6.4.2.3.1.2-1. ATCRBS CODE, RANGE, MONOPULSE, L/R AND M/S GENERATORS

The ARIES power attentuator has 1-dB steps. Zero attenuation corresponds to a -20 dBm SUM signal level referred to the Mode S sensor's RF port. We will assume that the output power level will be set to integral power P whenever the desired read power P_{ml} satisfies $(P + 0.5) > P_{ml} > (P - 0.5)$ dBm. P = -20 will be used for $-20 > P_{ml} > -20.5$ and P + -60 will be used for $-59.5 > P_{ml} \geq 60$. To find the corresponding range brackets, we invert the equation for P_{ml} to option:

$$r_{ml} = 10^{-}[1 + (P_{ml}/20)]$$

Therefore, the power should be set to P whenever:

$$10^{-}\{1 + [(P + 0.5)/20]\} < r_{ml} < 10^{-}\{1 + [(P - 0.5)/20]\}$$

Since r_{ml} is uniformly distributed between 1 and 100, the probability that it is in the above range is;

$$Pr_{ml}(P) = (1/99)*\{10^{-}\{1 + [(P - 0.5)/20]\} - 10^{-}\{1 + [(P + 0.5)/20]\}\}$$

$$= (1/990)*\{10^{-}[(P - 0.5)/20] - 10^{-}[(P + 0.5)/20]\}$$

for $(-21 > P > -59)$. Similar expressions apply for the end points P = -20 and P = -60.

If the same deviation is applied to r_{sl}, a similar expression for $Pr_{sl}(P)$ may be obtains as follows;

$$Pr_{sl}(P) = (1/310)*\{10^{-}[(P - 0.5)/20] - 10^{-}[(P + 0.5)/20]\}$$

for $(-21 > P > -49)$. Similar expressions apply for the end points P = -20 and P = -50.

Therefore, it is desirable to make the output of the ROM equal to P with the above probabilities. Since each ROM address is equally likely to be addressed, the way to do this is to have the number of ROM locations with P stored in them in each of the two distributions to be proportional to $Pr_{ml}(P)$ or $Pr_{sl}(P)$. Since there are 512 entries in the ROM, the number of entries containing P will, therefore, be 256 $Pr_{ml}(P)$ + 256 $Pr_{sl}(P)$. This number must be rounded up or down to obtain an integer number of entries.

6.4.2.3.1.3 Generation of Monopulse Angle Control Parameter.

The simulated fruit is characterized by a uniformly distributed monopulse offboresight angle. The offboresight angle control parameter is generated using an 11-bit random number generator, RNG5 (see figure 6.4.2.3.1.2-1). Its output drives a "Monopulse ROM" which is included in the present implementation, should it be necessary in the future to modify the distribution of this parameter. It serves no purpose at present.

6.4.2.3.1.4 Generation of Monopulse Angle Sign Control Parameter.

The monopulse angle sign parameter determines whether simulated fruit comes from the left or right of boresight. The parameter is a single bit, L/R bit, generated by random number generator No. 4, RNG4 (see figure 6.4.2.3.1.2-1). Note that this generator is also used in the generation of the range (power) control parameter. It provides for half of all replies being to the left of boresight and half to the right of boresight. The L/R parameter performs its function by switching a phase shifter in the analog portion of the FAT generator.

6.4.2.3.1.5 Generation of Random ATCRBS Codes.

There are 16 pulses in the ATCRBS reply. In the order of transmission they are:

$$F_1C_1A_1C_2A_2C_4A_4 \text{ X } B_1D_1B_2D_2B_4D_4F_2 \text{ SPI}$$

The F_1 and F_2 pulses are always present; ARIES assumes that the X and SPI pulses are always absent for fruit replies.

The distribution of the remaining bits depends on the type of interrogation causing the reply. For mode A (identity) interrogations, this assumes that discrete codes are much more likely than nondiscrete. This is not a correct model of any particular environment due to the fact that the A(=$A_4A_2A_1$) bits and B bits are chosen from a small set of codes assigned to any given control area. However, it is undesirable from a system testing point of view to model a particular environment that closely.

The mode C (altitude) code consists of two Gray codes, one encoding 100-foot increments and one encoding 500-foot increments. The C bits encode the 100-foot increments using the C values: 1, 2, 3, 4, and 6. These values are treated as being equally likely.

The A and B bits are used to encode altitude intervals which cover the typical altitude distribution of targets in such a way as to make all combinations almost equally likely. Exceptions are the A_1, and A_2 bits which are on for the intervals from 15,000 to 47,000 feet (A_1) and 7,000 to 23,000 and 39,000 to 55,000 feet (A_2), where the traffic densities are lower. ARIES ignores this factor and generates uniformly distributed A and B bit values.

The D bits are 0 for almost all mode C replies. The exception is that targets above 32,000 feet have the D_4 bit set. The D_2 bit is also a legal altitude bit, but is only set above 63,000 feet and so is not of interest. To approximate these conditions, it is assumed that approximately 15 percent of the replies are from targets above 32,000 feet, corresponding to a moderately high level of overflight traffic.

To generate uniformly distributed A and B bits for both altitude and identity codes, the 6 lower bits of a 15-bit random number generator are used, RNG1 (see figure 6.4.2.3.1.2-1). The remaining 9 bits are used to generate the C and D bits. Since the C and D bits have nonuniform distribution, a conversion ROM (the C, D code ROM), containing the proper distribution is included. Of the 512 ROM locations, 342 of them (about two-thirds) contain uniformly distributed values for C and D generated by simply counting through all values a sufficient number of times to fill the space. These provide the simulated data for mode A replies.

The remaining 170 ROM locations have C values uniformly distributed over the set 1, 2, 3, 4, and 6. These are again generated by successively counting through this set of values until the space is filled. Twenty-five of these locations have the D_4 bit set, the other have D = 0. The 6 bits from RNG1 and the 6 bits from C, D code ROM constitute the 12-bit random ATCRBS code.

The fraction of random ATCRBS codes generated (the rest have a fixed, know code) is determined by a fruit environment parameter provided by the computer.

This 5-bit ratio is loaded into the "% fixed code" register. With the LSB equal to 100/16 percent of fixed replies, the 5 bits can represent from 0 (no fixed replies) to 100 percent (16 = all fixed replies). Numbers above 16 are treated the same as 16.

To generate the proper percentage, the lower 4 bits of the "% fixed code" register are compared with 4 bits of the output of a 7-bit uniformly distributed random number generator, RNG2 (see figure 6.4.2.3.1.2-1). The results, as determined by the comparator, are used to select which of the two inputs to the multiplexer will be used as the ATCRBS code. When the output of RNG2 is smaller, the ATCRBS code generated will be used. Otherwise, the code stored into the "fixed code" register (part of the fruit environment parameters) will be used. The MSB of the "% fixed code" ratio is used to override the comparator. It is OR'ed with the output of the comparator. When this bit is set, all replies generated are fixed code.

6.4.2.3.1.6 Generation of the Delay-to-Trigger Control Parameter.

The delay to trigger control parameter determines when a fruit reply should be triggered by a fruit reply generator (FAT). To realistically simulate fruit, their arrivals as a function of time should be characterized by a Poisson or exponential distribution. The way in which the ARPG generates delay to trigger times with such a distribution by combining the outputs of two random number generators with uniform distribution is explained below.

The statistics of fruit arrivals are assumed to correspond to those of a Poisson process, and, therefore, the inter-arrival time is a random variable with an exponential probability density. The probability that the interval between two successive events is between "t" and "t + dt" is therefore;

$$P(t) = ye^{-yt}\,dt$$

where "y" is the fruit rate. In ARIES, the fruit rate must be variable over the range from 1,000 to 50,000 fruit/second in 1,000-fruit/second increments.

The approach taken is to approximate the exponential distribution by a step function which can be generated by combining the output of several random number generators with uniform distributions. Such a step function approximation is shown in figure 6.4.2.3.1.6-1, with only a small number of steps for the sake of clarity. In the actual implementation, 1024 steps are used.

To generate an exponentially distributed random number, a uniform number "n" between 1 and N is first generated. This is used to obtain t_n from a table (implemented in ROM). This essentially corresponds to picking one of the horizontal strips with probability 1/N, which matches the probability that each strip represents.

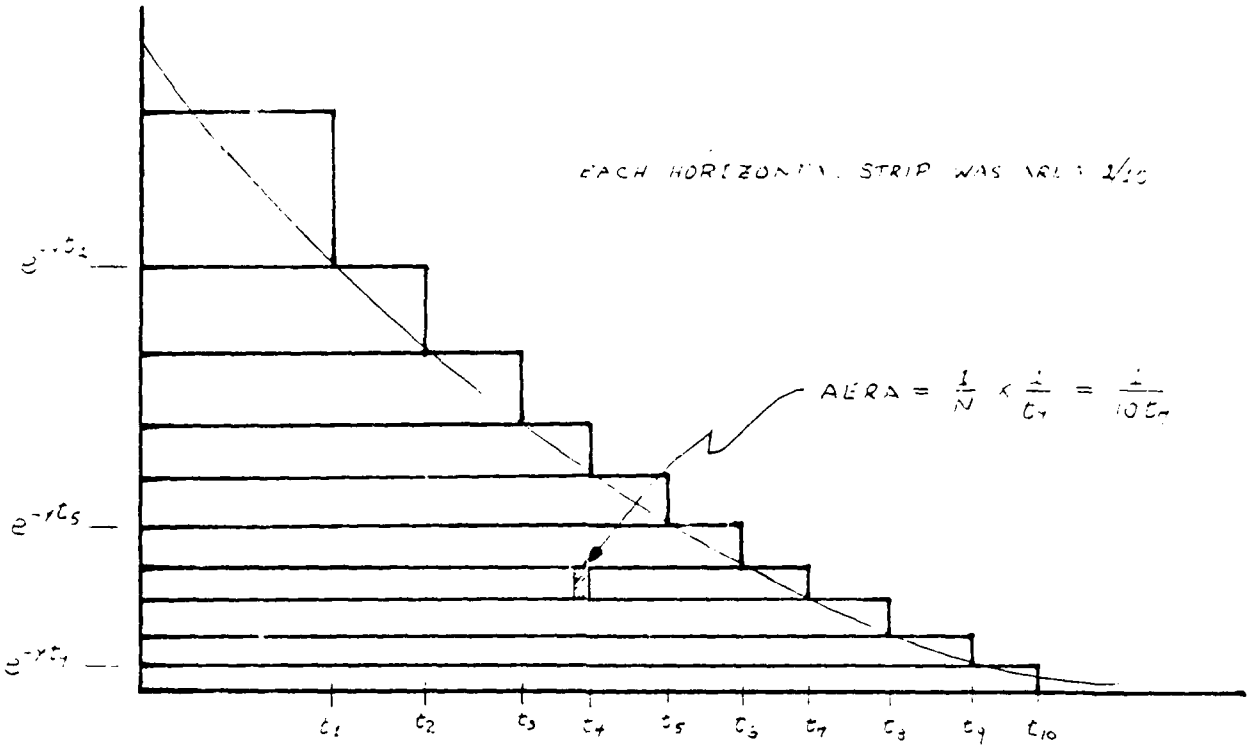


FIGURE 6.4.2.3.1.6-1. STEP FUNCTION APPROXIMATION TO EXPONENTIAL CURVE

Next, another uniform random number generator is used to obtain a number between 0 and t_n. The probability of picking a given number is $1/t_n$ (the t_n are integers). and the overall probability is $1/(N * t_n)$, representing a small area of probability, such as the filled in rectangle in figure 6.4.2.3.1.6-1. The total probability of picking any "t," corresponds to the sum of all such rectangles in the column for "t," and can be seen to be the desired step function.

To calculate the t_n, an expression for the area of each strip is set equal to 1/N, and the resulting equality solved for t_{n-1} in terms of t_n:

$$t_n \, [y \exp(-y * t_{n-1}) - y \exp(-y * t_n)] = 1/N$$

$$t_{n-1} = -(1/y) \ln[(1/N * y * t_n) + \exp(-y * t_n)]$$

The value t_n is set equal to the maximum value that can be stored in a lookup table, and then the above recursion is used to generate the other entities. To preserve numerial accuracy, all calculations must be performed in floating point or double precision and the resulting t_n then converted to integers of the desired size.

To provide for varying "y," the t_n stored in the lookup table are chosen for the maximum fruit rate, y_{max}, to be generated. Then, after some t_n is obtained, it is multiplied by y_{max}/y, where $y < y_{max}$ is the desired fruit rate. Then a uniformly distributed random number between 0 and $(t_n * y_{max})/y$ is generated.

This essentially corresponds to dilating all the inter-arrival times by the factor y_{max}/y. To see that $t'_n = (t_n * y_{max})/y$ are the correct values that would have been stored in a lookup table generated specifically for the desired "y," substitute "y" and $t'_n = (t_n * y_{max})/y$ in the above iteration. Then;

$$yt'_n = (y * t_n * y_{max})/y = t_n * y_{max}$$

and so the expression within the logarithm does not change value, and only the change in 1/y has an effect. Therefore, $t'_{n-1} = (t_n * y_{max})/y$ for all "n," by induction.

By this means all fruit rates $y < y_{max}$ can be generated from the lookup table for y_{max} simply by multiplying the t_n from the table by y_{max}/y. Note, however, that because of the limited precision of the t_n as actually stored, $(t_n * y_{max})/y$ may not be exactly the value that would have been calculated by using the recursion above with floating point arithmetic of greater precision. This is not a significant effect for ARIES' purpose.

Figure 6.4.2.3.1.6-2 is a block diagram of the overall Poisson sequence generator. The 15-bit random number generator RNG6 provides 10-bit uniformly distributed numbers. This is used to obtain an 11-bit value, t_n, from the Poisson distribution ROM. Then t_n is multiplied by a 12-bit scaling factor coming from the "fruit rate" register which is controlled by the computer. It is loaded into the "fruit rate" register as part of the fruit environment parameters from the FRC. The values of the scaling factor lie between 64 and 4096, corresponding to fruit rates of 64,000 to 1,000, respectively (scaling factor = 4096/desired fruit rate in thousands).

The output of the multiplier is a 23-bit number. The low order 6 bits are ignored, however, effectively dividing the result by 64. Therefore, the effective scaling factor lies between 1 and 64.

The resulting 17-bit number represents the upper bound, t_n, of the horizontal strip as discussed earlier (see figure 6.4.2.3.1.6-1). To choose a number from 0 to this upper bound, another uniform distributed random generator, RNG7, is used. This 17-bit generator is iterated and compared to t_n until a number less than or equal to t_n is found. For small t_n, this may take a significant amount of time, as most of the numbers generated by RNG7 are large. To get around this, the output of RNG7 is masked, eliminating most of the numbers greater than t_n.

The mask is generated by the Mask ROM. When the MSB of t has the value of 2^m then $(2^{m+1} - 1)$ is output from the mask ROM to the AND gates, allowing all values equal or smaller than $(2^{m+1} - 1)$ to get through.

Once a number is found, it is temporarily stored in the delay-time register. Before it is transferred into the FIFO, the 17-bit number must first be reduced to a 16-bit quantity. The 17 bits are initially required so that there will be a large enough range to hold the longest delay-to-trigger values generated at the

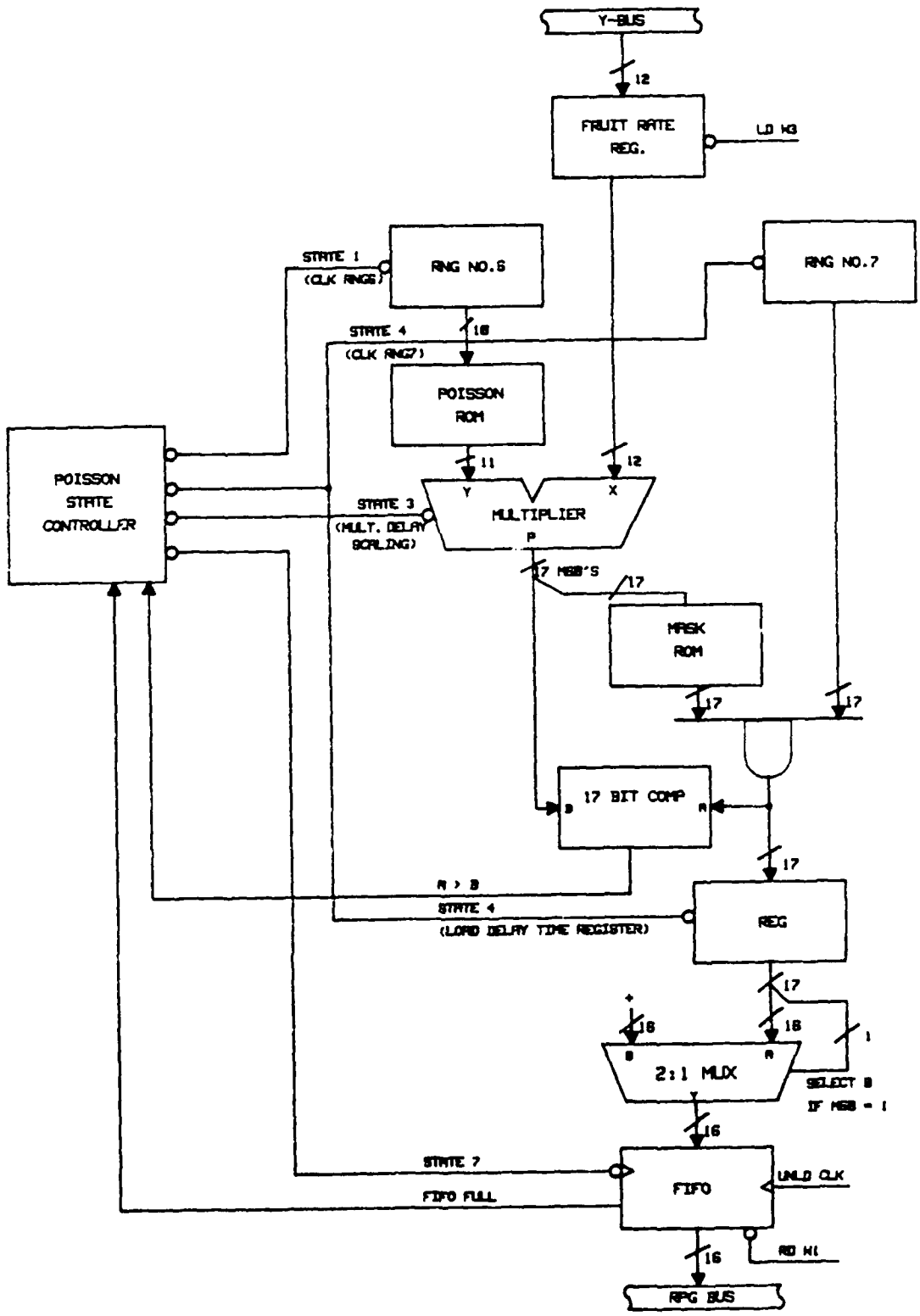


FIGURE 6.4.2.3.1.6-2. ATCRBS POISSON SEQUENCE GENERATOR BLOCK DIAGRAM

lowest fruit rates. To accomplish this, the MSB of the 17-bit number is used to select a multiplexer; otherwise, the lower 16 bits of the 17-bit number will be output. This scheme effectively reduces all numbers greater than $2^{16} - 1$ to $2^{16} - 1$, creating a truncated exponential distribution with a spike at the end.

At this point, if the ARPG is not currently being read by the FRC, the number will be immediately transferred to the buffer memory. The above process is repeated until the FIFO is full (contains 16 delay-to-trigger times).

This entire sequence of operations is controlled by the Poisson Sequence Controller. The state diagram for the controller is illustrated in figure 6.4.2.3.1.6-3. Not shown in the diagram is the reset input. When reset occurs, the Poisson Sequence Controller will return to state 0 from any state, and remain in this state until the reset is completed. As soon as reset is completed, the Poisson Sequence Controller will start generating 16 new delay-to-trigger times. State 6 is a waiting state, which is entered when both the FRC and the Poisson Sequence Controller try to access the FIFO at the same time. Reading by the FRC always has higher priority, therefore state 6 will remain until reading is over. At this time, a new delay-to-trigger time will be loaded into the FIFO (state 7).

Unloading of the FIFO by the FRC is discussed in the following section.

6.4.2.3.1.7 Reading the ARPG.

All six fruit control parameter generators discussed in the previous sections are tied to a common bus, called the RPG bus as shown in figure 6.4.2.3.1-1. To read the control parameters the FRC issues a read pulse, enabling the tristate output of each generator at an appropriate time, and allowing the parameters to be formatted in the output register as follows:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	DELAY-TO-TRIGGER TIME (LOWER 16 BITS)															
2	0	0	0	0	X	X	X	X	X	X	M/S	X	0	0	0	1
3	X	POWER						L/R	MONOPULSE							
4	ATCRBS CODE															

As each group of 16-bit words is output from the register, in four consecutive clock intervals, the FRC receives them via the D-Bus.

After the reading is completed, all random number generators, except RNG7, will be clocked once, allowing the generators to produce the next random numbers. The read control logic and its timing diagram are illustrated in figure 6.4.2.3.1.7-1.

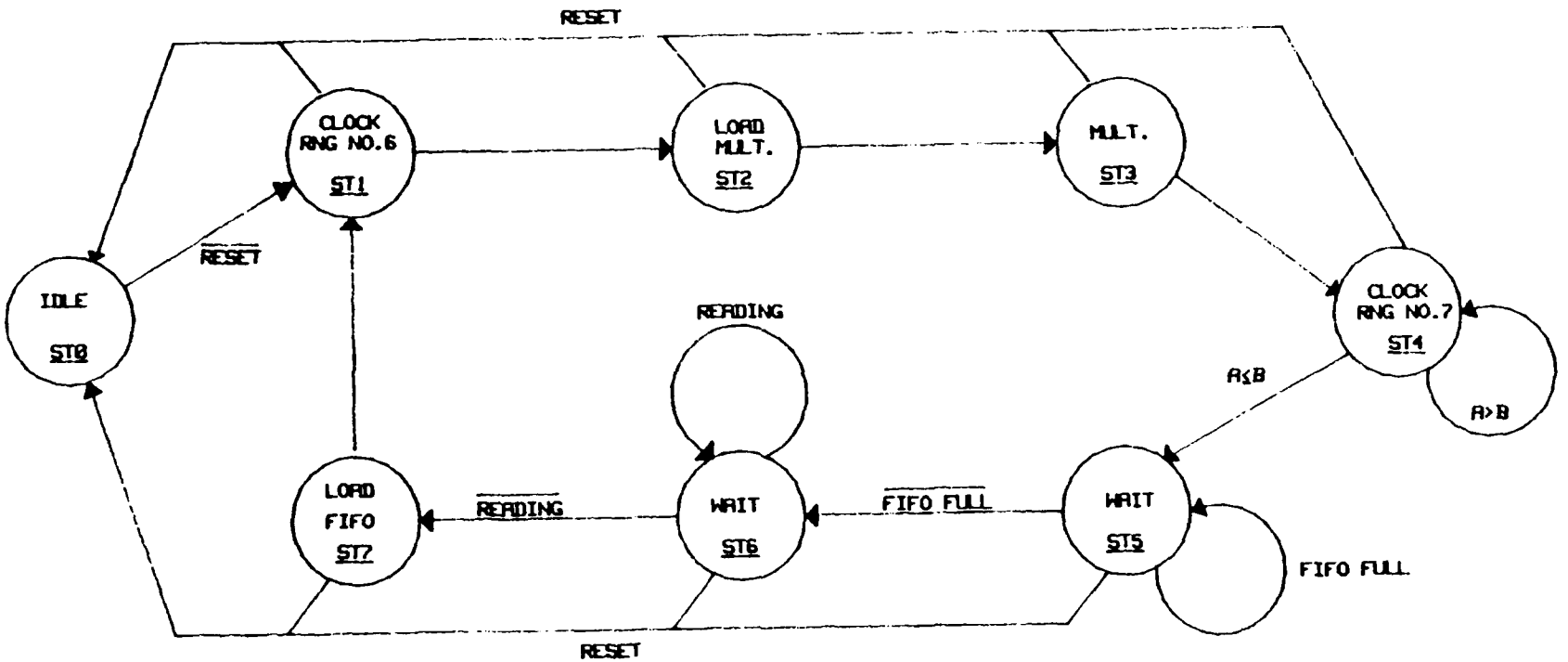


FIGURE 6.4.2.3.1.6-3. ATCRBS POISSON SEQUENCE CONTROLLER STATE DIAGRAM

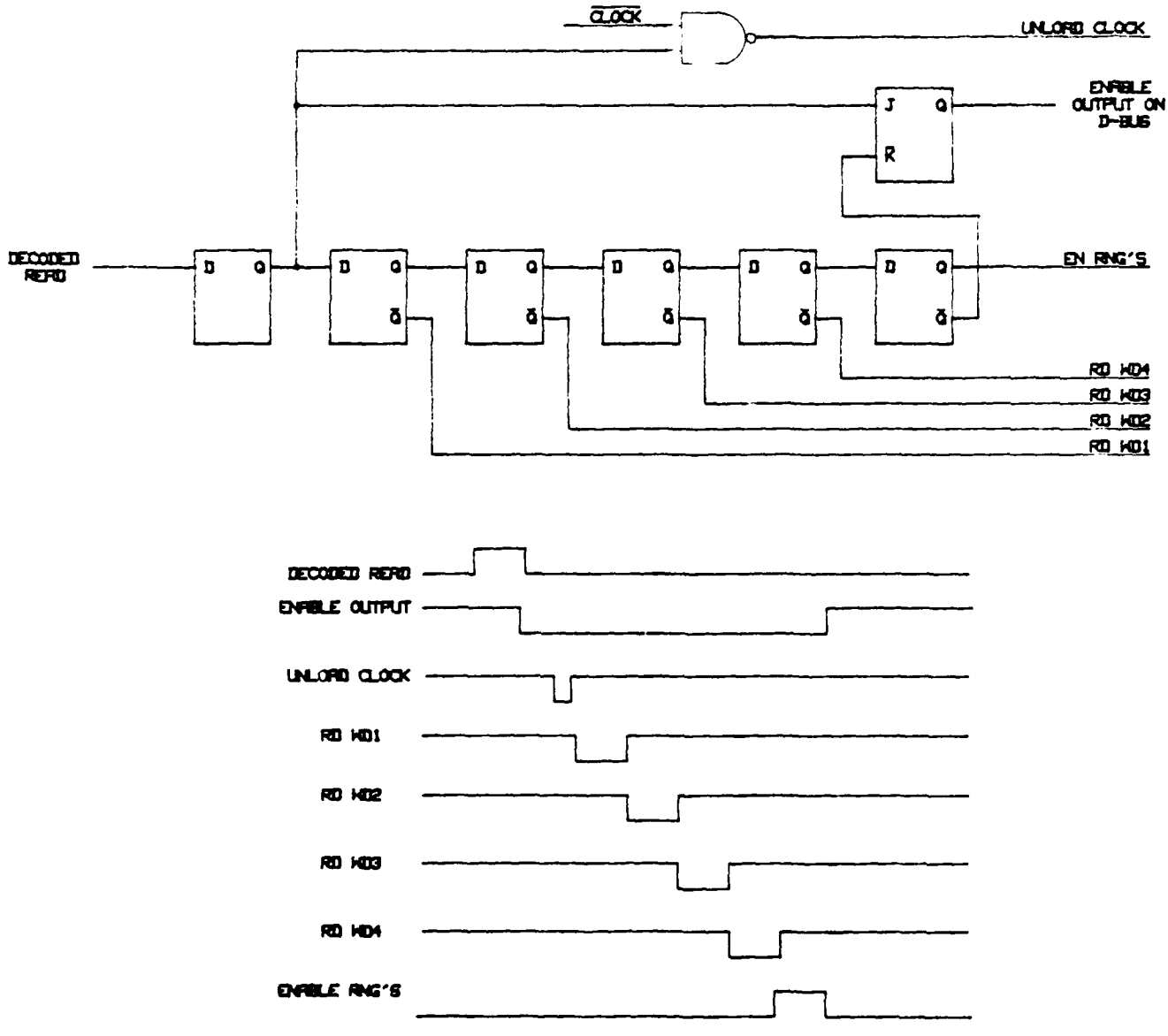


FIGURE 6.4.2.3.1.7-1. READ CONTROL LOGIC AND TIMING WAVEFORMS

6.4.2.3.2 Mode S Random Process Generator.

The MRPG constructs Mode S fruit reply blocks which are identical in format to those of the Mode S modeled reply blocks. These reply blocks are constructed with the use of three Mode S fruit environment parameters which are used to control the operation of the MRPG: (1) average fruit rate, (2) fraction of fruit in the mainbeam, and (3) fraction of long replies (as opposed to having short replies). These parameters are controlled, in the same manner as the ATCRBS fruit environment parameters, by the operational software via the FRC.

As shown in figure 6.4.2.3.2-1, the MRPG consists of six control parameter generators, each responsible for generating one of the above FAT control word parameters. Also shown are the interface circuits in which the control word groups are assembled and through which they are transferred from the MRPG to the FRC. Interconnections between the FRC and the MRPG are identical to those between the FRC and the ARPG. Control commands transmitted via the C-Bus from the FRC to the MRPG are:

RESET: Resets all random number generators to a predefined value, and generates 16 new inter-arrival times.

READ1-4: Causes the first 4 of 10 fruit control words to be output to the FRC.

READ5-10: Causes the last 6 of 10 fruit control words to be output to the FRC.

LOAD: Changes the average fruit rate, mainbeam/sidelobe ratio, and ratio of fixed code to random code.

6.4.2.3.2.1 Random Number Generators.

All of the pseudo-random number generators employed in the MRPG are identical in hardware to those of the ARPG except for one additional random number generator with 21 register stages (1 toggle flip-flop and 20 delay flip-flops). This generator is used in the generation of Mode S data and will be discussed in section 6.4.2.3.2.5. The reset sequence for the generators is four clock cycles longer than that of the ARPG to handle the larger generator. Other than that the reset circuitry is identical to the ARPG reset circuitry.

6.4.2.3.2.2 Generation of Random Range (Power) Control Parameter.

The hardware for the Random Range Generator is identical to that of the ARPG. (See section 6.4.2.3.1.2.)

6.4.2.3.2.3 Generation of Monopulse Angle Control Parameter.

The hardware for the Random Offboresight Angle Generator is identical to that of the ARPG. (See section 6.4.2.3.1.3.)

6.4.2.3.2.4 Generation of Monopulse Angle Sign Control Parameter.

The hardware for the Monopulse Angle Sign Generator is identical to that of the ARPG. (See section 6.4.2.3.1.4.)

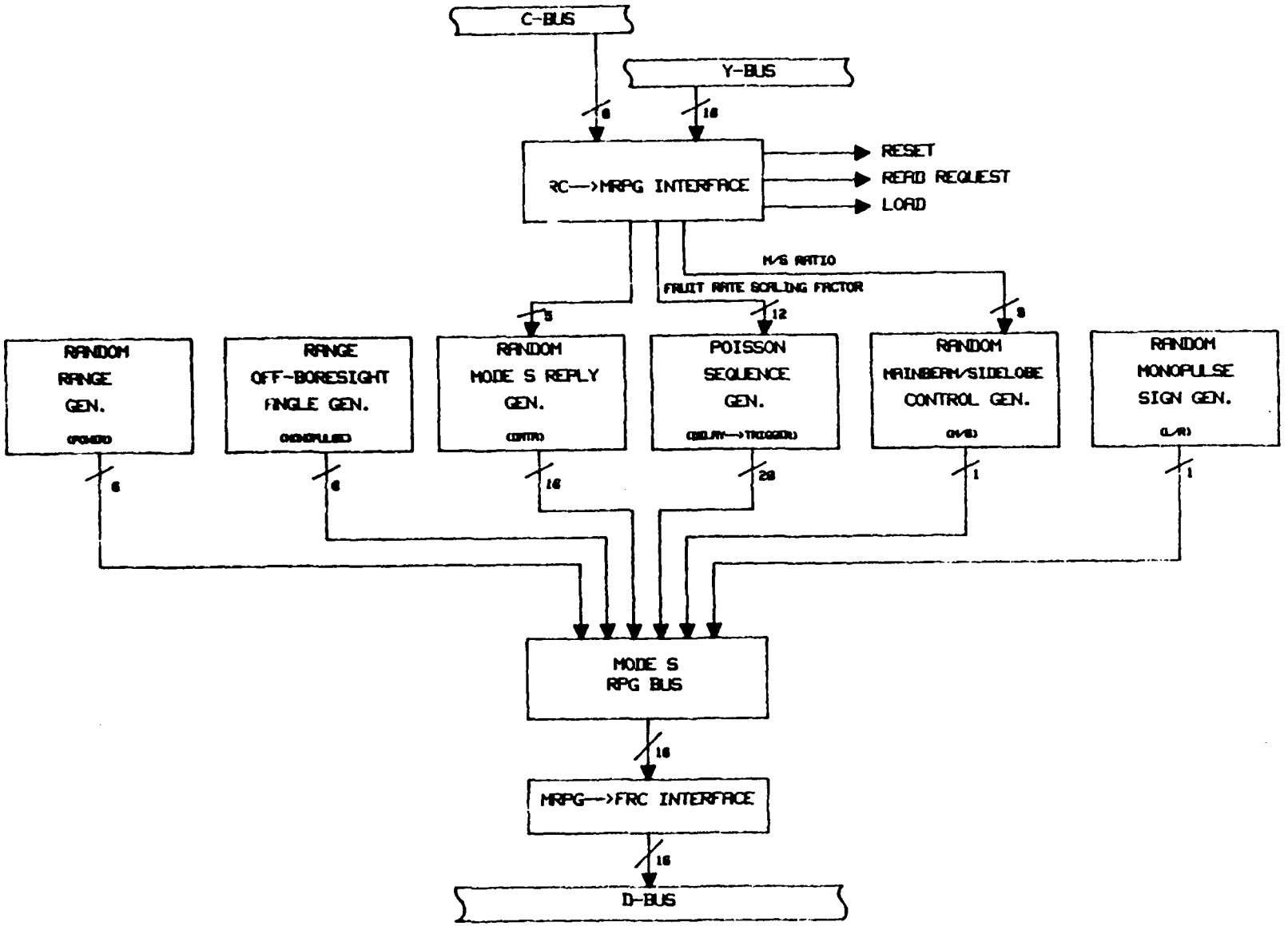


FIGURE 6.4.2.3.2-1. MRPG FUNCTIONAL BLOCK DIAGRAM

6.4.2.3.2.5 Generation of Mode S Data.

In order to simulate the Mode S fruit environment realistically, the type of replies most likely to be generated must first be considered. At present there are eight types of replies defined in the Mode S National Standard. Two replies, short and long special surveillance, are air-to-air reports. A third reply is the Comm-D or extended length message. These replies are not expected to be a significant percentage of the total Mode S beacon environment. Therefore, it can be assumed that the initial Mode S environment will essentially be made up of the following reply types and so the Mode S fruit environment must be made up of similar replies:

SURVEILLANCE (ALTITUDE)

SURVEILLANCE (IDENTIFICATION)

MODE S ALL-CALL

COMM-B (ALTITUDE)

COMM-B (IDENTIFICATION)

From the previous reply list, it is noted that the Mode S fruit environment will contain both short and long replies. The short replies may be surveillance type (altitude or identification) or Mode S All-Call. The long replies may be Comm-B (altitude or identification).

The fraction of long replies generated (the rest being short replies) is determined by a Mode S fruit environment parameter provided by the computer. This parameter, a 5-bit ratio is loaded into the "long/short reply" register. With the LSB equal to 100/16 percent of long replies, the 5 bits can represent from 0 (no long replies) to 100 percent (16 = all long replies). Numbers above 16 are treated the same as 16.

To generate the proper percentage, the lower 4 bits of the "long/short reply" register are compared with 4 bits of the output of a 7-bit uniformly distributed random number generator, RNG2, as shown in figure 6.4.2.3.2.5-1. The results of the comparater and the MSB of the "long/short reply" register are OR'ed to produce a single bit from the Random Long/Short (L/S) Control Generator that determines whether a reply will be long or short.

Once the types of replies to be simulated are determined, the data contents of the replies must be considered. Each reply contains a series of data bits (56 or 112) that are grouped into fields containing encoded information. Not all possible codes in some fields are defined and, therefore, not used. As an example, the 5-bit downlink format field has 8 of 32 possible codes defined at present. (This does not preclude that other codes will be defined in the future.) Also, the data fields may be unique between different reply types. Therefore, not only does the type of reply need to be considered but also what codes are possible in its data fields in order to simulate a realistic fruit reply.

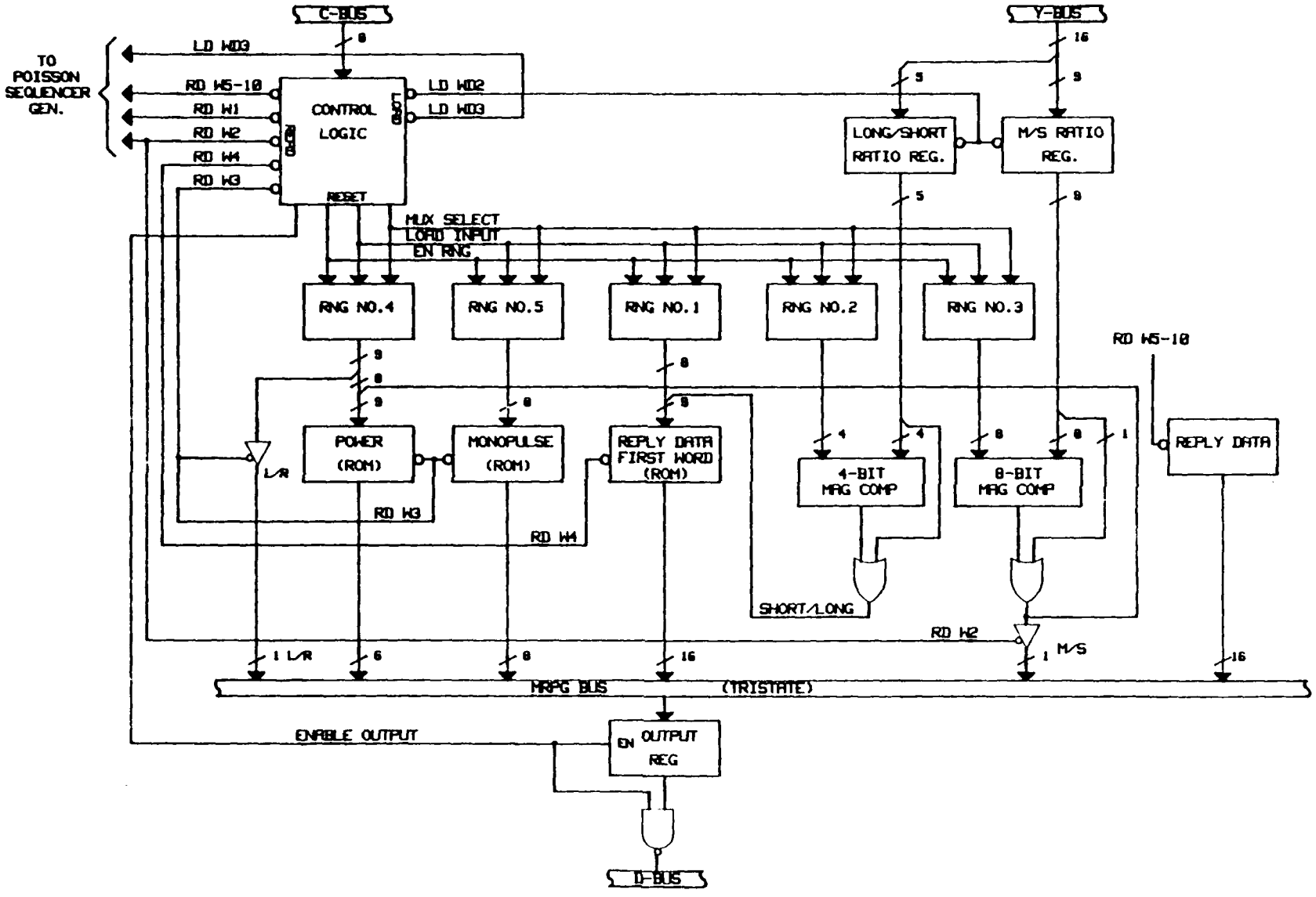


FIGURE 6.4.2.3.2.5-1. MODE S REPLY, RANGE, MONOPULSE, L/R AND M/S GENERATORS

As an example, there are five types of Mode S fruit replies that will be simulated. Each reply has a unique DF code that identifies the reply type and the information fields contained within the reply. Figure 6.4.2.3.2.5-2 illustrates the five Mode S replies, their information fields, and possible data patterns contained in their fields. The possible data patterns for each field are given equal probability of occurrence. The only modification to this is the way the altitude field is simulated which will be explained later.

Referring to figure 6.4.2.3.2.5-2, only the codes 0 through 5 are defined for the flight status field. Also in the downlink request field, only codes 0 and 1 are defined. In the case of the identification (ID) field, all of the bits have equal probability of occurrence. This is the same for the A, B, and C bits of the altitude code (AC) field. However, the D bits are 0 for almost all altitude replies as explained in section 6.4.2.3.1.5. The D_1 and D_2 bits are always absent in this case and the D_4 bit is present one-eighth of the time. The M bit position in the ID and AC fields is always absent. In the case of the Mode S All-Call reply, its format is significantly different from that of the other replies. In the capability field, only codes 0 through 3 are defined. Also, the Parity Identifier (PI) field contains 0. The remaining information fields can have any value with equal probability of occurrence.

The first Mode S word is generated using RNG1 and the Reply Data (first word) ROM shown in figure 6.4.2.3.2.5-1. The ROM is used to modify its uniform input distribution to provide a realistic data pattern for Mode S fruit replies. The contents of the ROM, as shown in figure 6.4.2.3.2.5-3, is distributed to give all defined codes in each field equal probability. The ninth address bit for the ROM is the L/S bit, and this is used to choose between two distributions. The first 256 locations contain the first word of a long Mode S reply, and the upper 256 locations contain the first word of a short Mode S reply.

The remaining Mode S words are generated by the circuitry as shown in figure 6.4.2.3.2.5-4. This circuitry consists of a 21-stage random number generator (RNG8), two 4:1-byte multiplexers (M_1 and M_2), each constructed from 4 dual 4:1 line multiplexer ICs, and controller logic. RNG8 provides uniformly distributed numbers to three of the inputs of M_1 and M_2. The random numbers are modified before being fed to two of the inputs of M_2 to simulate the data in the ID and AC fields more accurately. The fourth input of each multiplexer is grounded to provide 0 output when needed. Through this logic, the controller can construct at a byte level the remaining data to complete the Mode S reply.

The control logic is able to construct appropriate data to match the first word of the reply data by decoding three bits in the downlink format (DF) field of the first word. Based on these 3 bits, the controller selects the correct input of M_1 and M_2 to construct six words to complete the Mode S reply block. The first bit determines whether the reply is surveillance or Comm-B (see figure 6.4.2.3.2.5-2). The fifth bit determines whether the reply will contain the ID or AC field. The second bit, if set, overrides the other bits specifying that the reply is a Mode S All-Call.

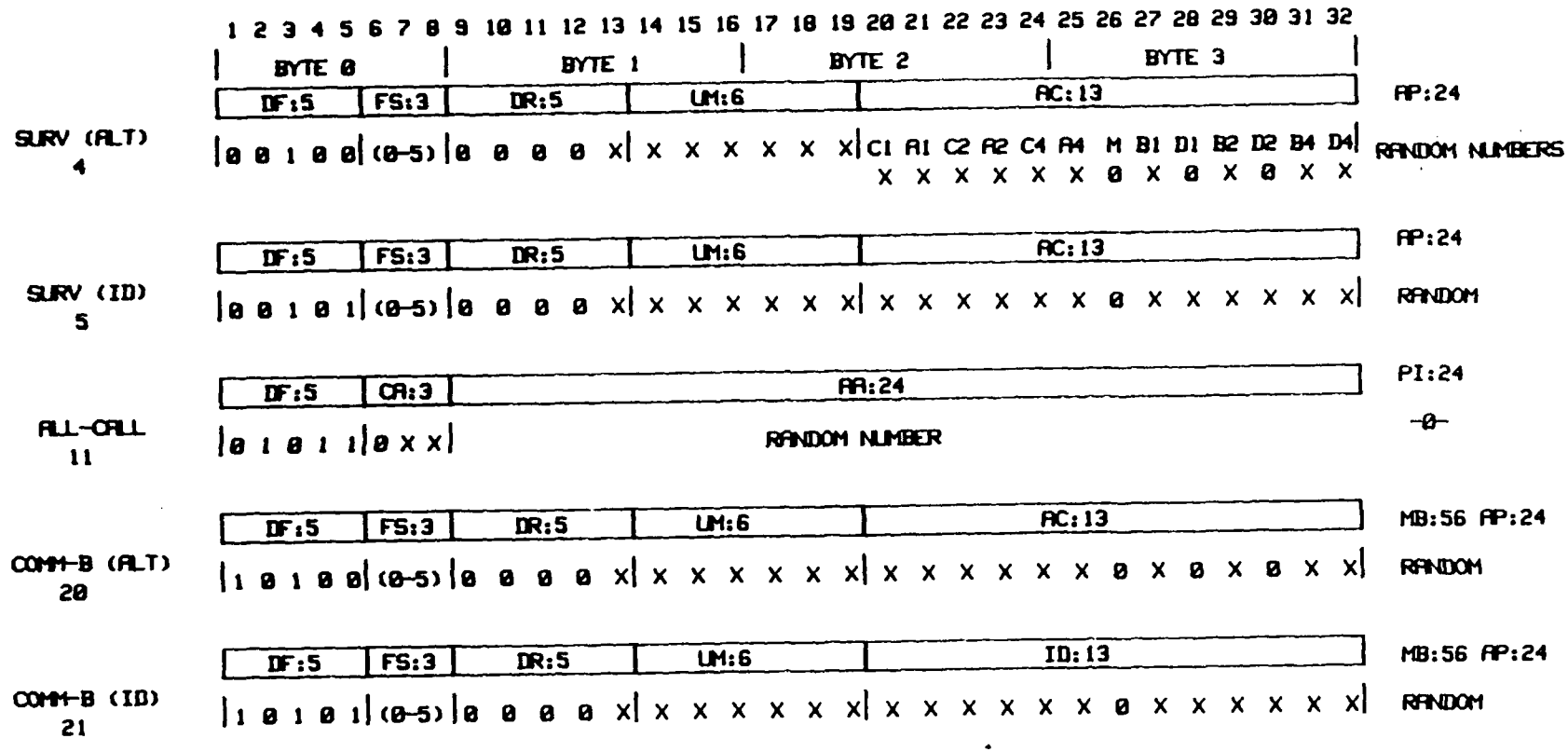
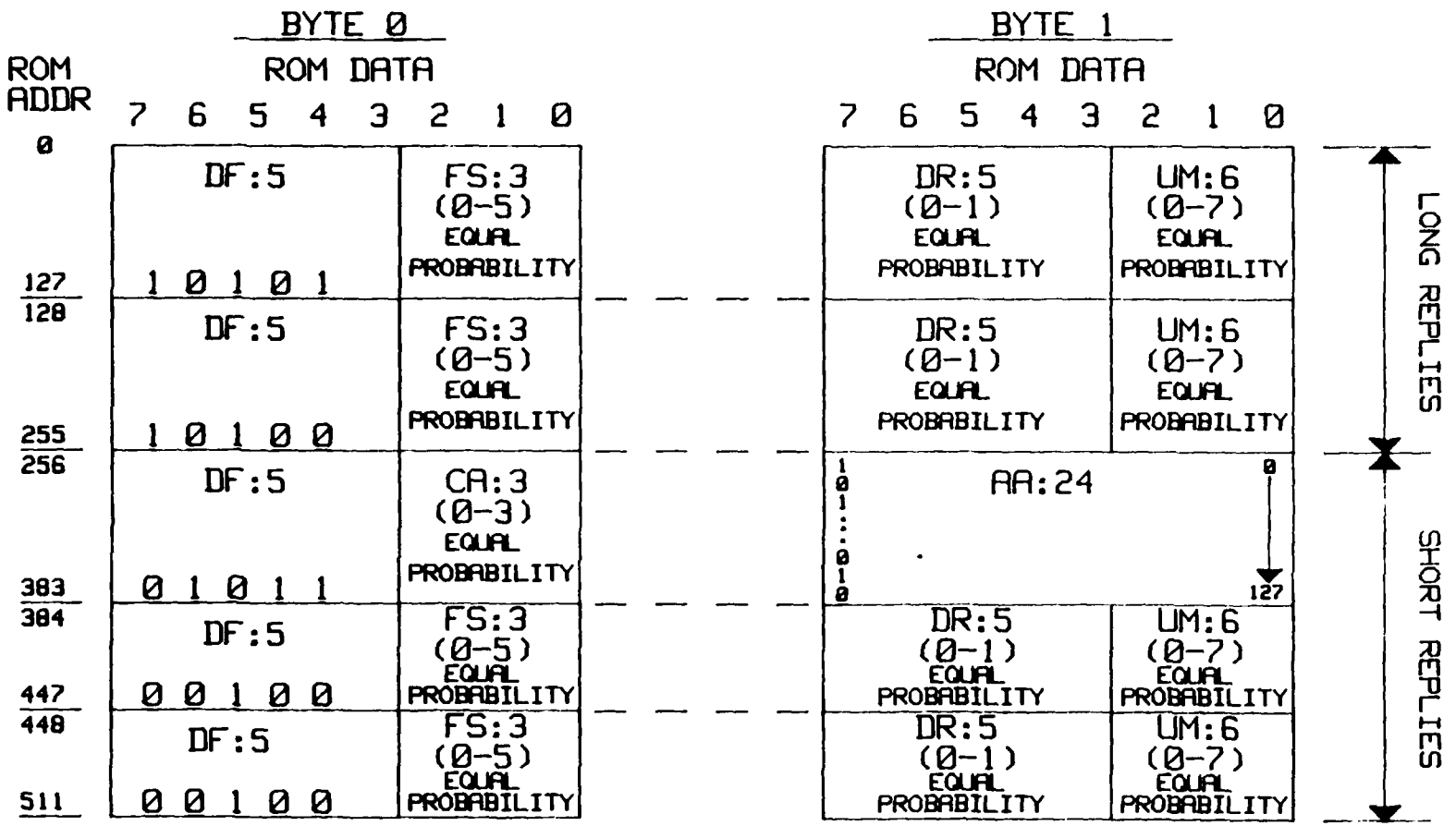


FIGURE 6.4.2.3.2.5-2. MODE S REPLY INFORMATION FIELDS AND POSSIBLE DATA PATTERNS



NOTES: 1. RNG N0.1 PROVIDES ADDRESS BITS A0 THROUGH A7 WHERE A0-A7 IS A UNIFORMLY DISTRIBUTED NUMBER BETWEEN 0 AND 255.

2. SHORT/LONG REPLY BIT CONTROLS ADDRESS BIT A8

FIGURE 6.4.2.3.2.5-3. DATA DISTRIBUTION FOR THE FIRST WORD OF A MODE S REPLY

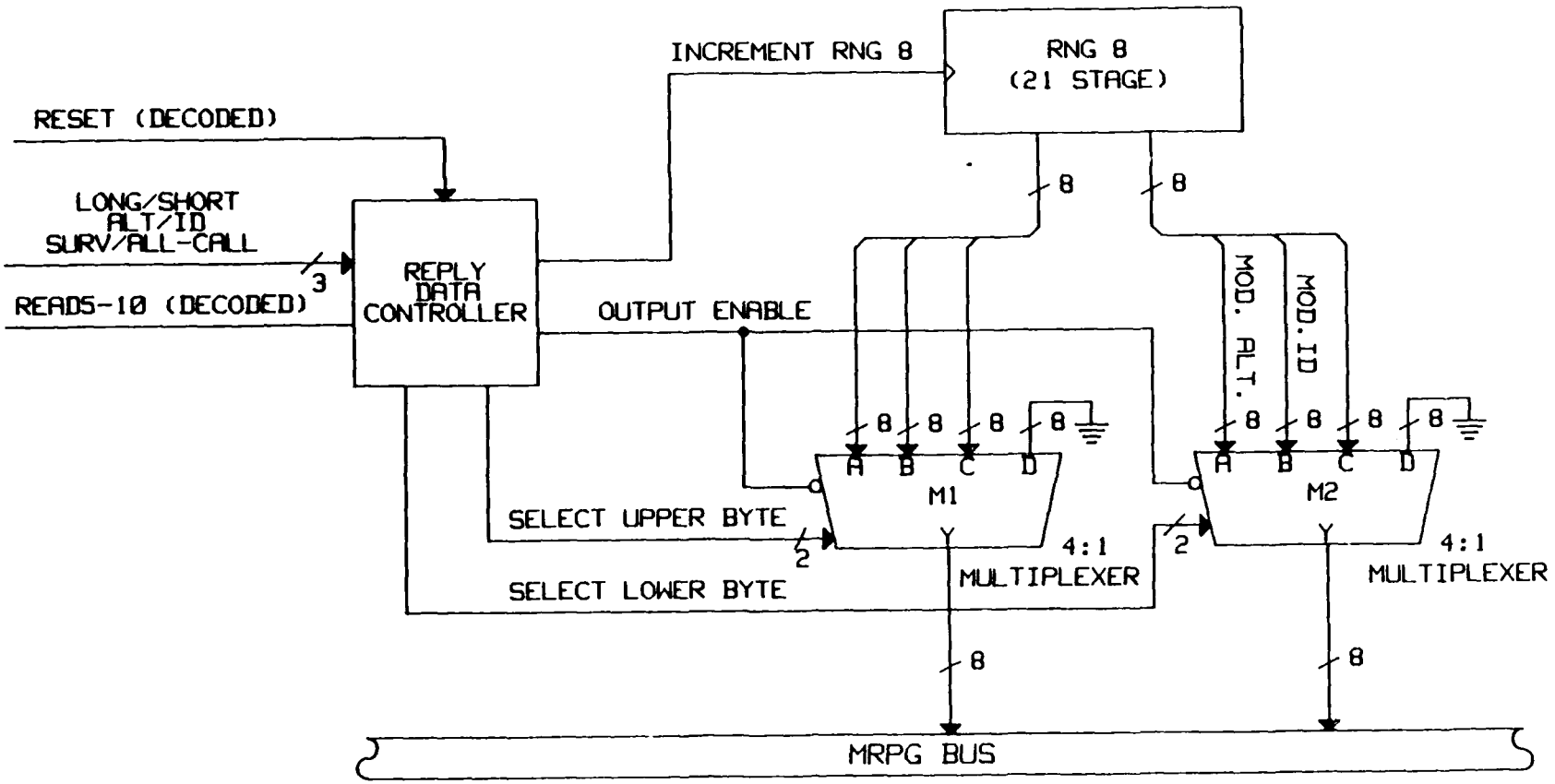


FIGURE 6.4.2.3.2.5-4. MODE S DATA GENERATOR BLOCK DIAGRAM

The three DF bits are latched into register REG_1 when the first reply data word is read as shown in figure 6.4.2.3.2.5-5. These bits are fed to address lines of a ROM to select the construction sequence that will build the appropriate data to match the first reply word. The controller does not begin the sequence until a READ5-10 command is decoded at which time J/K flip-flop IC_1 is set. Once the sequence begins six data words are generated. Each time a word is read, RNG8 is incremented to a new value to generate the next word. Flip-flop IC_1 is reset after all six data words are read.

6.4.2.3.2.6 Generation of the Delay-to-Trigger Control Parameter.

Functionally, the Mode S Poisson Sequence Generator is identical to that of the ATCRBS Poisson Sequence Generator. It provides inter-arrival times according to an exponential distribution to generate fruit rates over the range of 10- to 500-fruit/second in 10-fruit/second increments. These rates are calculated using the same formulas given in section 6.4.2.3.1.6.

Modeling the fruit rate inter-arrival times according to Poisson statistics at low fruit rates of 10-fruit/second, the probability of inter-arrival times of 6 to 7 seconds cannot be ignored. To achieve time durations this long, running at a range clock rate of 16 MHz, the inter-arrival time field would need to be 23 bits wide. However, only 20 bits are allocated in the reply message block giving a maximum achievable inter-arrival time slightly greater than 1 second at the normal range rate. Therefore, to increase the inter-arrival time, the normal range clock (16 MHz) for the Mode S FAT was replaced with the 4-MHz clock essentially adding 2 bits to the inter-arrival time field. At this rate, inter-arrival times greater than 4 seconds are possible.

Referring to figure 6.4.2.3.2.6-1, the Poisson curve used by the MRPG is stored in ROM as a 2048-step function approximation. Each step has equal probability of being selected by RNG6 that provides a uniformly distributed number to the address lines of the ROM. Some values in the step function curve are less than 64 and are rejected by the sequence controller. This is because an inter-arrival time of 0 would be produced if a number less than 64 is multiplied by the scaling factor and the least significant 5 bits of the result are truncated. The rejection of numbers less than 64 is accomplished by having one bit from the Poisson ROM fed back to the Sequence Controller. If this bit is set high, flagging that the value selected is less than 64, state 1 is repeated. RNG6 is incremented. State 1 will be repeated until this bit goes low indicating that a step selected from the exponential curve is greater than or equal to 64. This is the only difference between the state diagrams of the ARPG and MRPG Poisson State Controllers.

6.4.2.4 Fruit Reply Generator Interface.

The FRC Interface:

a. Provides fruit parameters to the ARPG and the MRPG via the FRC

b. Supports diagnostic tests on both RPGs

c. Supports diagnostic loopback tests on the four FATs

d. Supports diagnostic loopback tests on the FRC

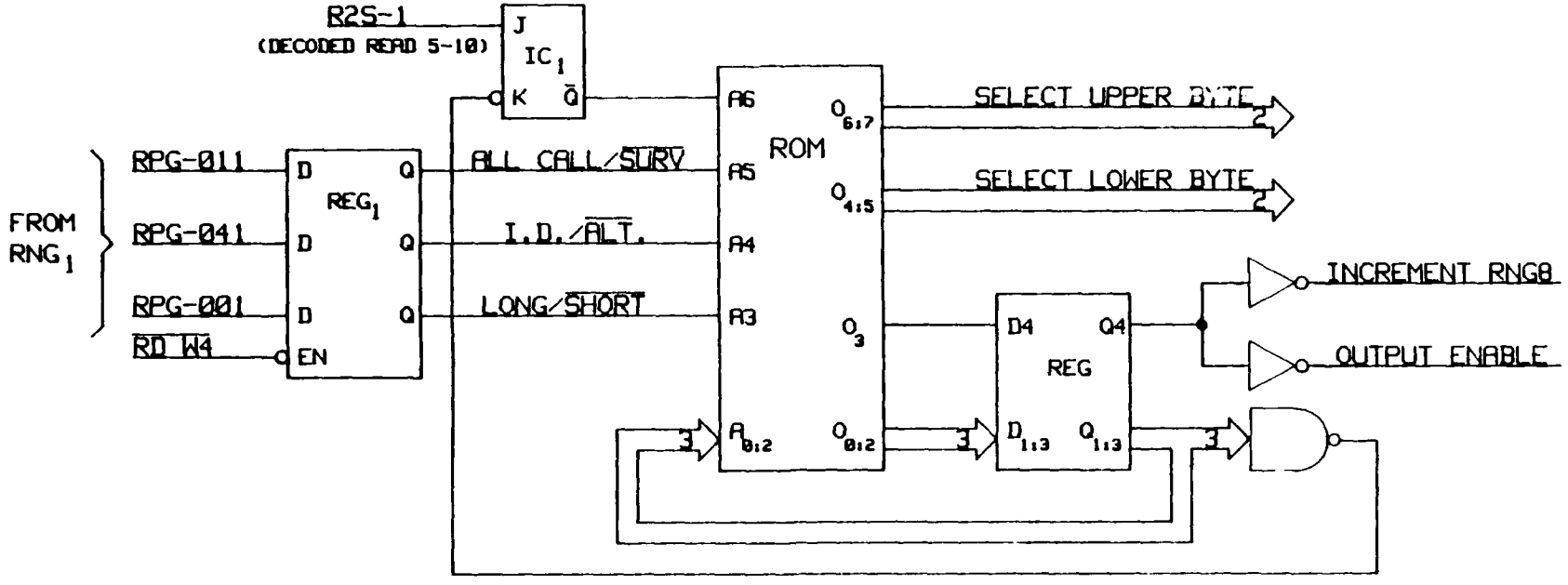


FIGURE 6.4.2.3.2.5-5. REPLY DATA CONTROLLER LOGIC DIAGRAM

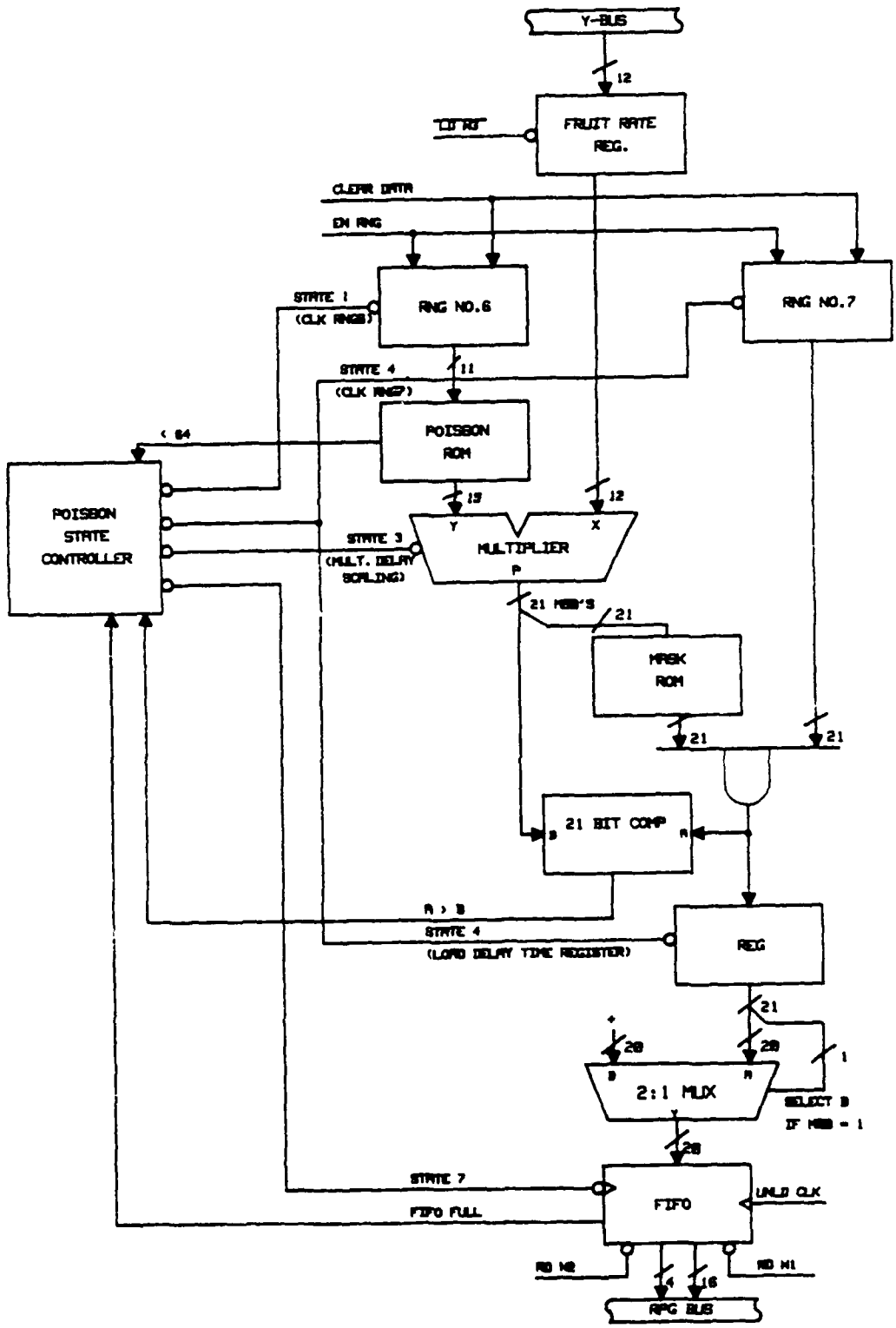


FIGURE 6.4.2.3.2.6-1. MODE S POISSON SEQUENCE GENERATOR BLOCK DIAGRAM

e. Supports diagnostic loopback tests on the Interface itself

f. Supports diagnostic tests is conjunction with the STU on the analog circuitry of the FATs

The FRG Interface, figure 6.4.2.4-1, contains three nearly independent channels, two of which are used for transferring fruit environment parameters and reply data blocks from the computer to the FRC, and the remaining channel transfers in the opposite direction. Each channel consists of 16-bit x 16-word memory, a memory address counter, and each uses 1 bit to indicate whether the memory is in a read or write mode. The memories used for these channels are labeled the PARA-RAM, the RPLY-RAM, and the CPU-RAM, respectively.

In the normal mode of operation, and in some of the diagnostic modes, fruit environment parameters are passed from the computer to the FRC via the PARA-RAM. This is accomplished by the circuitry shown in figure 6.4.2.4-2. Note that the output of the address counter $CNTR_1$ is wired to the memory for locations 0 to 7 with the MSB address bit of the memory fixed to ground. This results in 8 out of 16 possible locations of the memory being used. This is necessary since all transfer cycles must be in groups of eight words (four words allocated for ATCRBS fruit environment and four words allocated for Mode S fruit environment).

To begin a transfer, provided that the PARA-RAM circuitry is enabled by the mode decoding logic to be discussed later, the computer first checks the status bit to be sure that it is allowed to write into memory. If the status bit value is 1, the computer has access to the memory. After eight words have been written into memory, the Q_D output of $CNTR_1$ goes high and is lead-edge detected to produce a pulse to change to the PARA-RAM status flip-flop in the opposite state. This signals the completion of the write cycle and allows the FRC to read from memory. At this point $CNTR_1$ contains a value of 8. However, since the Q_D output is not used for addressing, the address is left pointing at the first word stored (location 0).

The status bit is coupled to a 16-bit gate array constructed of four quad open-collector NAND gate ICs that represents the interface status word. When the gate array is enabled via a decoded status request sent by the FRC, the status word is placed on the D-Bus enabling the FRC to examine the read/write mode of the PARA-RAM circuitry. The status flip-flop will set back to the write mode when the FRC finishes reading eight words.

The reading cycle is performed by having the FRC execute two separate four-word read cycles. This is necessary since the first four words to be read must be sent to the ARPG and the last four words read must be sent to the MRPG. When a READ1-4 command is decoded by the Controller Command Decode logic, counter $CNTR_2$ is preset to a count of 12. The Q_D output of $CNTR_2$ is coupled back to its P-EN input allowing it to increment until it loops back to 0 at which time Q_D goes low inhibiting the counter. During this period, the Q_D output is high for four clock cycles enabling the address counter $CNTR_1$ via OR gate A to step through address locations 0 to 3 and stop at 4. During the same interval, the Q_D output line is used to place the parameter words on the D-Bus so they may be read by the FRC. When the second READ1-4 command is decoded, the previous read sequence is repeated, this time stepping through address locations 4 through 7 and stopping at 0. Actually, $CNTR_1$ counts to 15 and wraps back to 0. When the overflow signal is generated, it is fed to the status flip-flop setting it back to the write mode completing the read cycle.

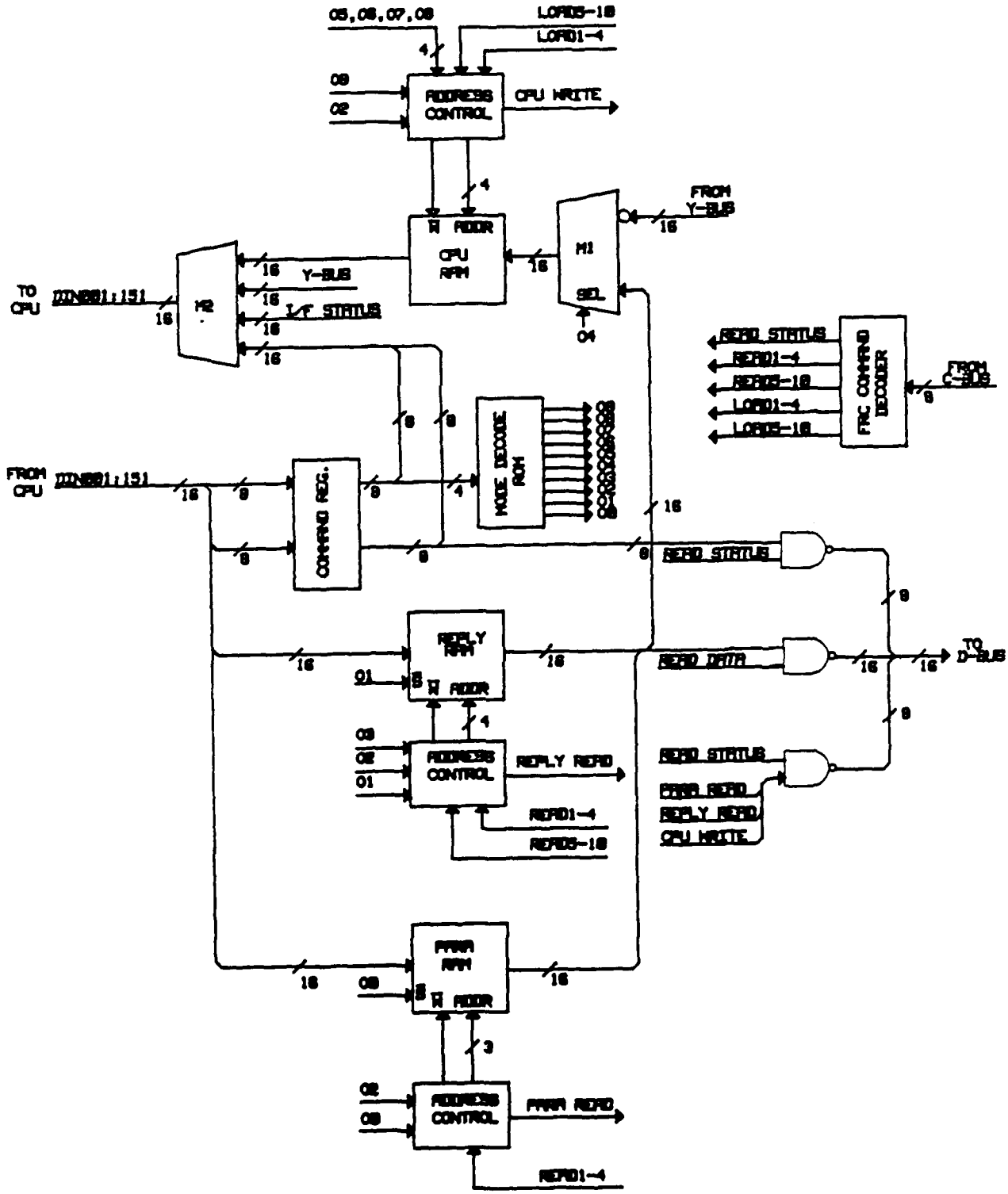


FIGURE 6.4.2.4-1. FRG INTERFACE BLOCK DIAGRAM

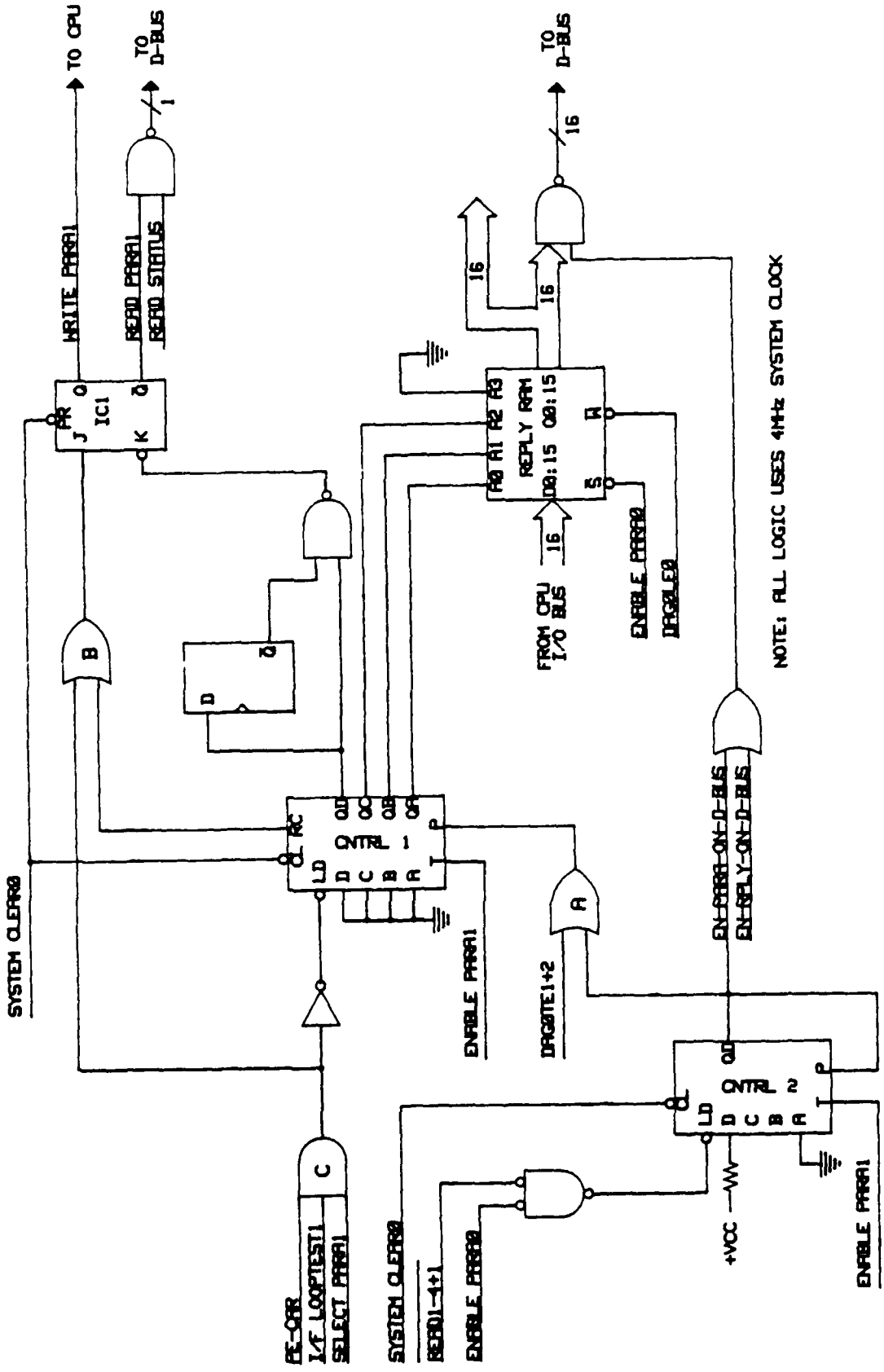


FIGURE 6.4.2.4-2. PARAMETER INPUT CHANNEL CIRCUITRY

Note that the address counter $CNTR_1$ and the status flip-flop IC_1 are initialized in order that the computer can access the memory whenever a hardware reset is executed. AND gate C and OR gate B are used to support special interface loopback diagnostic modes to be covered in section 6.4.2.4.1.7.

In diagnostic support modes, the RPLY-RAM is used to pass either 4- or 10-word data blocks from the computer to the FRC. This is accomplished by the circuitry shown in figure 6.4.2.4-3. Note that the address counter $CNTR_1$ is wired so that it can only count from 11 to 15 or 5 to 15, resulting in only 4 or 10 out of 16 possible locations in memory being used. This is necessary since all transfer cycles must be in groups of four, the size of ATCRBS reply block message, or in groups of ten, the size of a Mode S reply block message.

To begin a transfer, provided that the RPLY-RAM circuitry is enabled by the Mode Decoding logic, the computer first checks the status bit to be sure that it is allowed to write into memory. If the status bit value is 1, the computer has access to the memory. Depending on the ATCRBS/MODE S bit from the mode decoding logic, the read/write cycles will be 4 or 10 words long. Once the specified number of words have been written, the overflow signal generated by $CNTR_1$ toggles the RPLY-RAM status flip-flop IC_1 in the opposite state and presets $CNTR_1$ to point to the first word stored. This signals the completion of the write cycle and allows the FRC to read from memory.

The RPLY-RAM status bit is fed to the interface status word to allow the FRC to examine the read/write state of the RPLY-RAM circuitry. The status flip-flop will set back to the write mode automatically when the FRC finishes reading.

The reading cycle is performed by having the FRC execute one of two possible read cycles. This is necessary to support transfers of 4 or 10 words. When a READ1-4 command is decoded by the Controller Command Decode logic, counter $CNTR_2$ is preset to a count of 12. The Q_D output of $CNTR_2$ is coupled back to its P-EN input allowing it to increment until it loops back to 0 at which time Q_D goes low inhibiting the counter. During this period, the Q_D output is high for 4 clock cycles enabling the address counter $CNTR_1$ via OR gate A to step through locations 11 to 15 if 4 words were written or through locations 5 to 9 if 10 words were written. During the same interval, the Q_D output line of $CNTR_2$ is used to place the data on the D-Bus so they may be read by the FRC. When a READ5-10 command is decoded, $CNTR_2$ is preset to 10 and its Q_D output remains high for six clock cycles incrementing the address counter $CNTR_1$ through locations 9 to 15 to access the remaining six words of a 10-word transfer. When the overflow signal is generated by $CNTR_1$, the status flip-flop is set back to its write mode and $CNTR_1$ is reloaded with the preset value.

Note that the RPLY-RAM status flip-flop IC_1 is initialized so that the computer can access the memory whenever a hardware reset is executed. Also, that the address counter $CNTR_1$ is cleared to 0 which is not one of the starting locations. Therefore, it is necessary to preload the starting address for the proper transfer cycle before writing begins. This is accomplished by software specifying the data transfer size and generating a preload pulse to the address counter through NOR gate C. AND gate D is used to support special interface loopback diagnostic modes to be covered in section 6.4.2.4.1.7.

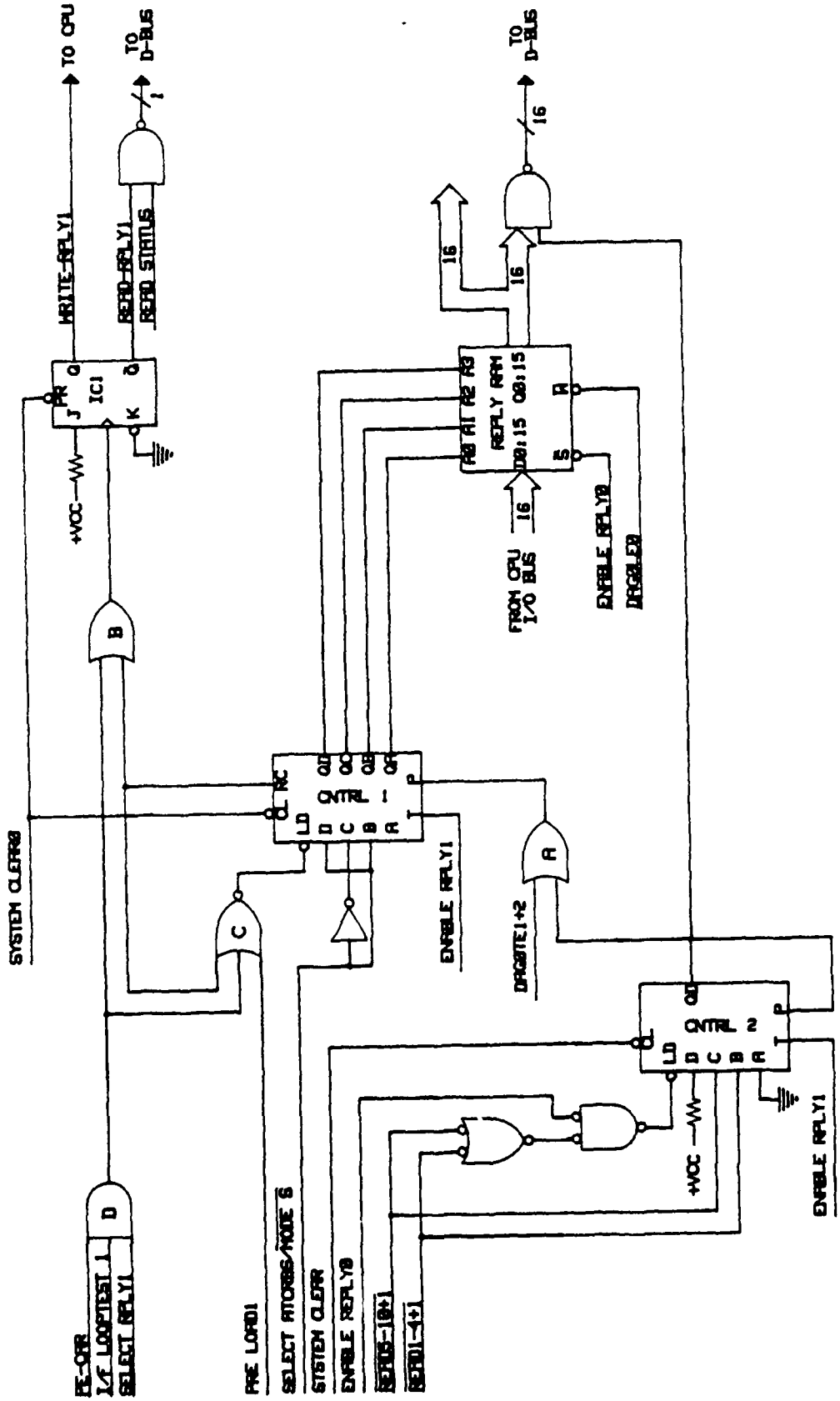


FIGURE 6.4.2.4-3. REPLY BLOCK INPUT CHANNEL CIRCUITRY

In diagnostic operations data are returned to the computer from the FRC via the CPU-RAM. This is accomplished by the circuitry shown in figure 6.4.2.4-4. This circuitry is very similar to the RPLY-RAM circuitry and operates much in the same way.

For the FRC to begin a transfer, provided that the CPU-RAM circuitry is enabled by the Mode Decoding logic, the read/write status bit for the CPU-RAM is checked. If the status bit is in the write mode, the FRC is allowed to access memory. Once the specified number of words have been written, the overflow signal generated from address counter $CNTR_1$ points to the CPU-RAM status flip-flop IC_1 to the read mode and presets $CNTR_1$ to the first word stored. This signals the completion of the write cycle and allows the computer to read from memory.

The computer reads data from the CPU-RAM by issuing the data request instruction once it has been determined that the CPU-RAM is in the read mode. After each data request, the address counter $CNTR_1$ is incremented to the next location. When the last word is read, the CPU-RAM status flip-flop IC_1 is toggled back to the write mode and $CNTR_1$ is preset back to the first location.

Note that the CPU-RAM status flip-flop IC_1 is initialized so that the FRC can access the memory whenever a hardware reset is executed. Also, that the address counter $CNTR_1$ is cleared to 0 which is not one of the starting locations. Therefore, it is necessary to preload the starting address for the proper transfer cycle before writing begins. This is accomplished by a 4-bit preload address field taken from the Mode Decoding logic to specify the data transfer size. This is necessary since several data transfer sizes are possible (4-, 8-, and 10-word transfers) and depends on the diagnostic mode selected. NOR gate A and AND gate B are used to support special interface loopback diagnostic modes to be covered in section 6.4.2.4.1.7.

Data is coupled to the CPU-RAM via multiplexer M_1, constructed of four 2:1 quad multiplexer ICs. The Y-Bus is connected to one input of M_1. A tristate bus shared by the PARA-RAM and the RPLY-RAM to pass data on the D-Bus is connected to the other input of M_1. This provides a means of loop testing the interface, as the data are sent to one of the input channels (PARA-RAM or RPLY-RAM) by the computer. The data can then be verified by reading it back from the output channel (CPU-RAM). Input that is passed through M_1 depends on the diagnostic mode selected.

The memory output is fed to another multiplexer M_1 (see figure 6.4.2.4-1), constructed of eight 4:1 dual multiplexer ICs. The Interface status, the FRC status, and the command instruction are fed to the other inputs of M_2. The output of M_2 is the input data bus for the computer. This multiplexer provides the means for the computer to check the Interface hardware, to examine the status of the Interface and the FRC, or retrieve data from the FRC. Two bits from the command instruction are used to select which input is placed on the input data bus (refer to section 6.4.2.4.1, Mode Decoding logic for further details).

6.4.2.4.1 Mode Decoding Logic.

As mentioned previously, the Interface operates in many different modes. A mode is selected by setting the mode field of the Command instruction as specified in table 6.4.2.4.1-1. The format for the FRC command instruction is given in appendix D of the ARIES Hardware Maintenance Manual, under the heading Fruit Reply Generator.

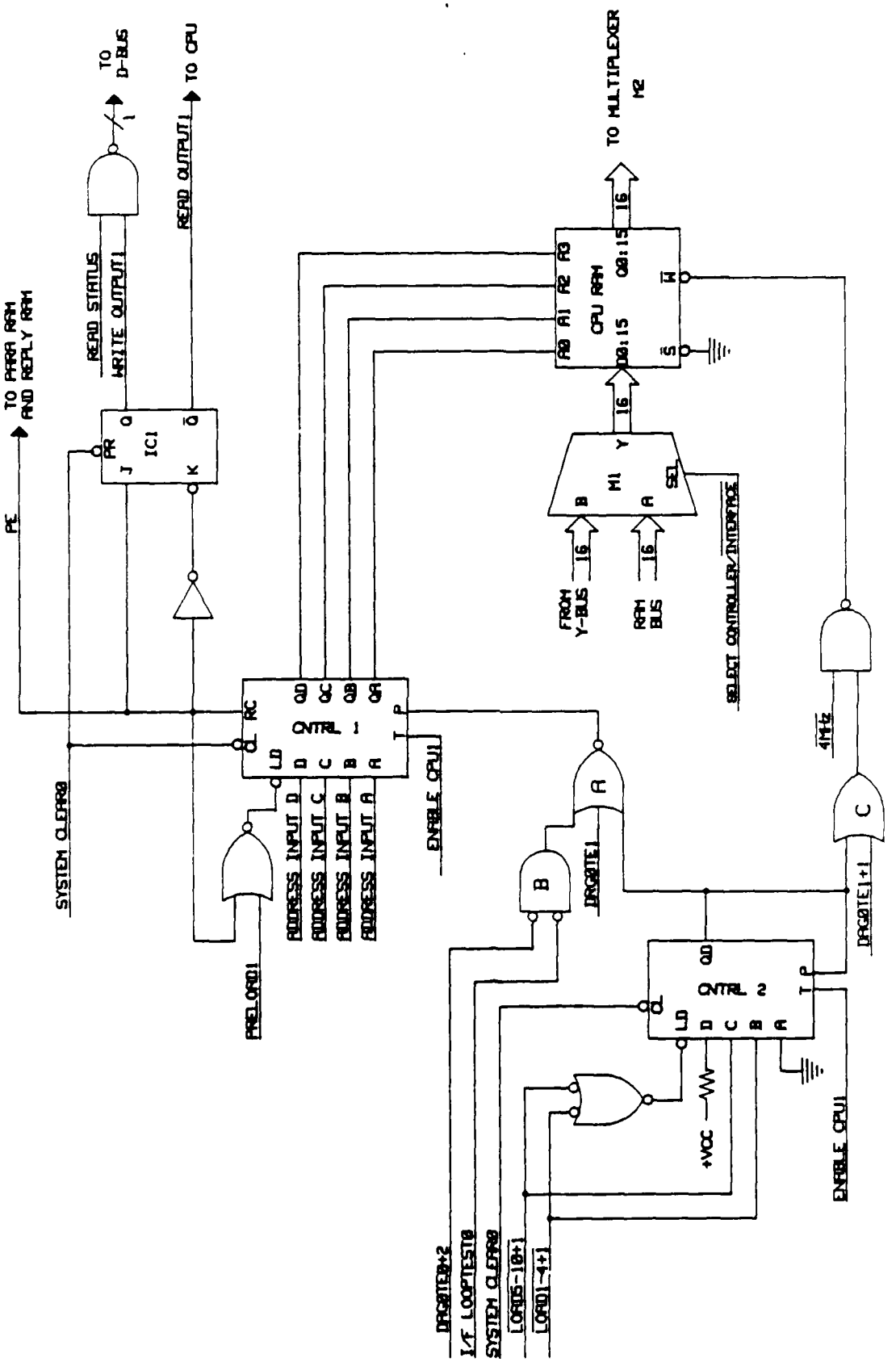


FIGURE 6.4.2.4-4. OUTPUT CHANNEL CIRCUITRY

Figure 6.4.2.4.1-1 shows the Mode Decoding logic used in the FRG Interface. This logic consists of a two-level command register and a mode decoding ROM. The first register REG_1 is used to latch 16-bit instructions from the computer. The second register REG_2 in used to synchronize the instruction to the 4-MHz system clock before it is passed to other circuitry. Four of the bits from REG_2 are connected to the address lines of the Mode Decode ROM. These bits form the mode field. This field is decoded by the ROM to set up the configuration control bits to enable the proper address counter(s), select the proper data paths, and enable the proper memory write pulses for each mode of operation.

In addition, it is necessary to pass the mode of operation to the FRC. This is accomplished using 3 bits of the ROM to form the operations control field (see table 6.4.2.4.1-1). This field is examined by the FRC when the interface status word is read.

TABLE 6.4.2.4.1-1. MODE DECODE CONFIGURATOR ROM

MODE FIELD OP CODE	OPERATING MODE	FRC OP CONTROL FIELD	* INTERFACE CONFIGURATION CONTROL BITS
			O_9 O_8 O_7 O_6 O_5 O_4 O_3 O_2 O_1 O_0
0	NORMAL	0 0 0	0 0 0 0 0 0 0 0 0 1
1	PARA-RAM DIAG	1 1 1	1 0 1 1 1 0 0 1 0 1
2	ATCRBS-RAM DIAG	1 1 1	1 1 0 1 1 0 1 1 1 0
3	MODE S-RAM DIAG	1 1 1	1 0 1 0 1 0 0 1 1 0
4	ARPG DIAG	0 1 1	1 1 0 1 1 1 0 0 0 1
5	MRPG DIAG	1 0 0	1 0 1 0 1 1 0 0 0 1
6	FAT DIAG	0 1 0	1 1 0 1 1 1 1 0 1 0
7	ATCRBS RF DIAG	0 0 1	0 0 0 0 0 0 1 0 1 0
8	MODE S RF DIAG	0 0 1	0 0 0 0 0 0 0 0 1 0
9	FRC DIAG	1 0 1	1 1 0 1 1 1 1 0 1 0

* O_9 - ENABLE CPU-RAM
O_8 - INPUT ADDRESS D
O_7 - INPUT ADDRESS C
O_6 - INPUT ADDRESS B
O_5 - INPUT ADDRESS A
O_4 - SELECT CONTROLLER/INTERFACE
O_3 - SELECT ATCRBS/MODE S
O_2 - INTERFACE DIAG MODE
O_1 - ENABLE RPLY-RAM
O_0 - ENABLE PARA-RAM

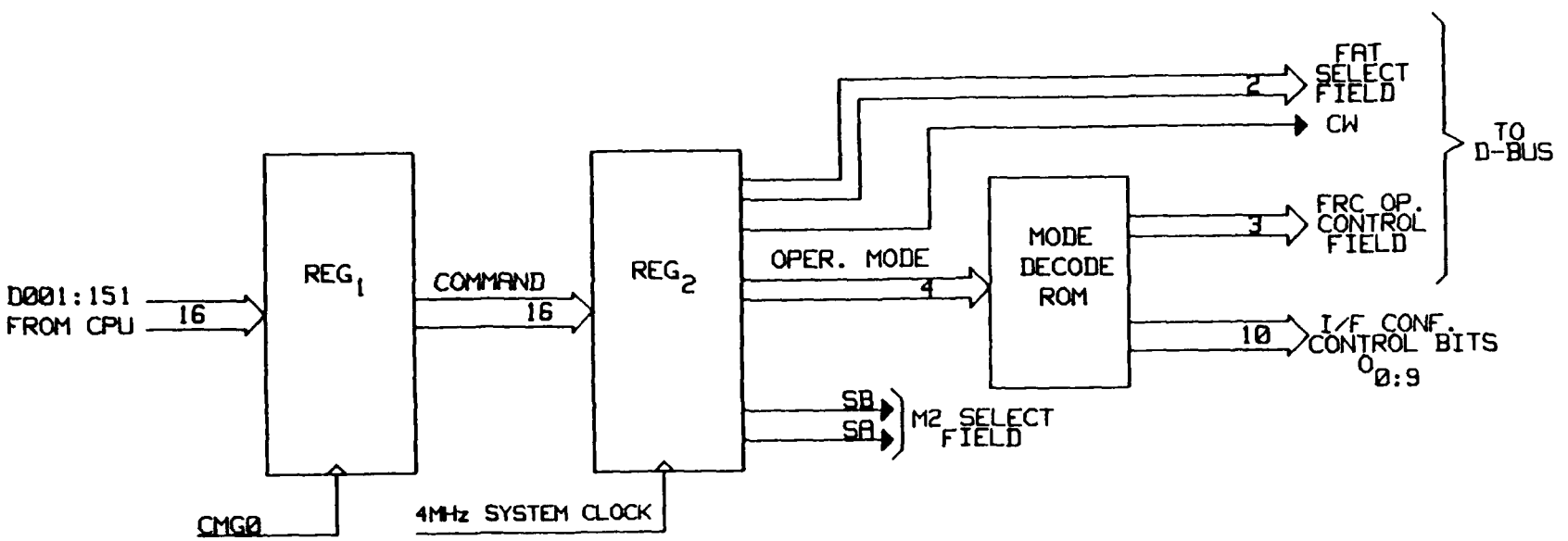


FIGURE 6.4.2.4.1-1. INTERFACE MODE DECODING LOGIC

6.4.2.4.1.1 Normal Mode.

The normal mode, the usual operating mode of the Interface uses only the PARA-RAM circuitry and allows the computer to output fruit environment parameters to both RPGs via the FRC. The computer outputs these parameters to the PARA-RAM by means of eight DA0 instructions. Each DA0 is leading and trailing edge detected, creating two pulses DAG0LE0 and DRG0TE1+2, respectively. The number +2 seen after the second pulse specifies that the pulse was delayed two clock cycles after it was generated. The reason for edge detecting the DA0 (or DR0) instruction is twofold:

a. It is necessary to synchronize the instruction from the computer to the 4-MHz system clock.

b. Edge detected pulses are 1-clock period wide (250 ns) so that the address counter will increment only once for each DA0 (or DR0).

As shown in figure 6.4.2.4-2, DAC0LE0 is used to write the data into the memory and DRG0TE1+2 is used to increment the memory address counter after each write.

6.4.2.4.1.2 ARPG Diagnostic Mode.

The ARPG diagnostic mode provides a way of checking the ARPG operation using the computer. Both the PARA-RAM and the CPU-RAM are used. To begin the operation, the computer initializes the RPGs by sending fruit environment parameters to them, using the PARA-RAM, exactly the same as in the Normal Mode. The FRC then requests a fruit reply from the ARPG and sends it to the computer using the CPU-RAM. This operation continues until enough data are gathered to check the ARPG.

To write four words to the CPU-RAM, the FRC issued a LOAD1-4 command. This command is decoded by the FRC Command Decode logic to produce a pulse LD1-4+1, causing the write counter $CNTR_2$ (see figure 6.4.2.4-4) to produce four memory write pulses and one 4-clock wide pulse to the address counter $CNTR_1$. To read four words from the CPU-RAM, the computer issues four DR0 instructions to the Interface. As in the case of DA0 instruction, each DR0 is trail-edge detected forming the DRG0TE0 pulse. This pulse is used to increment $CNTR_1$.

6.4.2.4.1.3 MRPG Diagnostic Mode.

The MRPG diagnostic mode provides a way of checking the MRPG operation using the computer. Both the PARA-RAM and the CPU-RAM are used. To begin the operation, the computer initializes the RPGs by sending fruit environment parameters to them using the PARA-RAM, exactly the same as in the normal mode. The FRC then requests a fruit reply from the MRPG and sends it to the computer via the CPU-RAM. This operation continues until enough data is gathered to check the MRPG.

The four-word write cycle to the CPU-RAM is the same as described in section 6.4.2.4.1.2. To write six words to the CPU-RAM, the FRC issues a LOAD5-10 command. This command is decoded by the FRC Command Decode logic to produce a pulse LD5-10+1, causing the write counter $CNTR_2$ (see figure 6.4.2.4-4) to produce six memory write pulses and one 6-clock wide pulse to the address counter $CNTR_1$. To read 10 words from the CPU-RAM, the computer issues 10 DR0 instructions to the Interface.

6.4.2.4.1.4 Controller Mode.

The Controller mode provides a way of checking the FRC operations using the computer. Both the RPLY-RAM and the CPU-RAM are used. A four-word data block is loaded into the RPLY-RAM. The FRC then fetches the block from the RPLY-RAM and returns it to the computer via the CPU-RAM. This is repeated until enough data are gathered to check the FRC. Under this test mode, the data can be any pattern allowing complete checkout of the data buses (D-Bus, Y-Bus) between the Interface and the FRC. All remaining tests using the FRC require the data blocks to be in the correct reply format.

6.4.2.4.1.5 FAT Diagnostic Mode.

Normally, the FATs receive fruit reply information from the RPGs, via the FRC, and generate the appropriate modulation pulses to the analog circuitry to produce a reply at 1090 MHz. This mode allows the computer to insert known reply data into the FATs, bypassing the RPGs. Both the RPLY-RAM and the CPU-RAM are used.

To begin the operation, the computer loads an ATCRBS reply block (Mode S reply generation is not performed under this diagnostic mode) in the RPLY-RAM. The FRC then fetches the reply from the RPLY-RAM and transfers it to a previously selected FAT. The FAT is selected from the 2-bit FAT Selection field which is part of the computer command instruction. The controller then triggers the FAT to transmit the reply. As the reply is transmitted, diagnostic support logic integrated in the FAT samples the parameter control output lines from the Hold-2 Register and serially samples the code train as it is being transmitted. The sampled data is reconstructed into a four-word data block identical in format to that of a reply parameter message block. Once the data block is assembled, the FRC fetches the block and returns it to the computer via the CPU-RAM. This way, all of the ATCRBS functions of the FATs can be verified.

6.4.2.4.1.6 FRG Analog Mode.

The FRG Analog mode provides a way of checking the reply generator analog circuitry with the support of the STU. Only the RPLY-RAM is used during this mode and no data are returned to the computer via the CPU-RAM. Both ATCRBS and Mode S reply generation are performed under this diagnostic mode. Therefore, Mode S functions not checked in the FAT Diagnostic mode can be checked here.

6.4.2.4.1.7 Interface Loopback Mode.

The Loopback mode provides a means of loop testing the Interface since the data sent by the computer to either input channel can be verified by reading it back from the output channel. The FRC does not participate in this mode.

Data may be looped through the PARA-RAM or the RPLY-RAM. In the case of the RPLY-RAM, the data may be loaded as 4- or 10-word blocks. The Mode Decode logic automatically configures the Interface to support the selected mode. Since all three modes are similar except in the number of words transferred, only the four-word transfer through the RPLY-RAM will be described in detail.

Assume that the Interface was initialized and a command instruction was issued programming the Interface for the ATCRBS-RAM Diagnostic mode. At this point, the RPLY-RAM and the CPU-RAM would be enabled and the output of the RPLY-RAM would be fed to the CPU-RAM through multiplexer M_1 (refer to figure 6.4.2.4-1). The correct address values would be loaded into both address counters set up by the Mode Decode logic. Also generated by the Mode Decode logic, special logic gates are enabled by the control signal I/F Looptest to support the Interface Loopback modes.

To the computer, writing and reading operations are similar to the other modes involving four DA0's and DR0's, respectively. However, under this mode, each DA0 causes data from the RPLY-RAM to be written into the CPU-RAM. This is accomplished by the use of gates B and C in the Output Channel Circuitry (see figure 6.4.2.4-4). As noted earlier, data are loaded into the RPLY-RAM on the leading edge pulse (DAG0LE0) of a data available instruction and the address of the RPLY-RAM is not updated until two clock periods after the trailing edge pulse (DAG0TE1+2) is generated on the same instruction. Between this interval, the data loaded into the RPLY-RAM are placed at the input of the CPU-RAM via multiplexer M_1. The data are loaded into the CPU-RAM by the use of gate C which feeds the one clocked delayed trailing edge pulse (DAG0TE1+1) to the write enable of the RAM. On the following clock cycle, the CPU-RAM address is stepped to the next location by the use of gate B at the same time the RPLY-RAM address is updated.

This sequence is repeated until four words are loaded into the CPU-RAM. After the fourth word is loaded, the ripple carry is generated by $CNTR_1$ switching the CPU-RAM status flip-flop from the write mode to the read mode. At the same time, the ripple carry is fed back to the RPLY-RAM circuitry through gate D (see figure 6.4.2.4-3) to gates B and C. However, this signal does not effect the normal operation of the RPLY-RAM circuitry at this time because a ripple carry signal, generated by $CNTR_1$ of the RPLY-RAM circuitry, is fed to the same gates B and C at the same time. The RPLY-RAM status flip-flop is switched to the read mode preventing the computer from writing any further data to the RPLY-RAM.

When the computer reads four words from the CPU-RAM, the ripple carry signal is generated again by $CNTR_1$ switching the CPU-RAM status flip-flop back to the write mode preventing the computer from reading any further data. Again, the ripple carry is fed back to the RPLY-RAM circuitry through gate D to gates B and C. The RPLY-RAM status flip-flop is switched back to the write mode bypassing the normal read sequence performed by the FRC. Note at this time no ripple carry signal is generated from $CNTR_1$ of the RPLY-RAM. At this point, the computer can continue and write another group of four words into the RPLY-RAM. This way all of the data paths and data block transfers, required by the three RAMs, can be verified.

6.4.2.4.2 FRC Command Decode Logic.

The FRC Command Decode Logic shown in figure 6.4.2.4.2-1 consists of an 8-bit register, a 4-bit magnitude comparator, and a 3:8 line decoder. The register is used to realign commands on the C-Bus to the 4-MHz system clock before it is passed to other circuitry. The upper 4 bits of the command, containing the device address, are fed to one side of the comparator. The input on the other side of the comparator is set to the FRG Interface address. If the device address matches the address of the Interface, the 3:8 line decoder is conditionally enabled. The lower 4 bits of the command, containing the operation code, are fed to the 3:8 line decoder. The upper fourth bit is connected to a second chip enable input. If this bit is low, the 3:8 line decoder is enabled and decodes the lower 3 bits for the operating instruction. These instructions are listed in table 6.4.2.4.2-1.

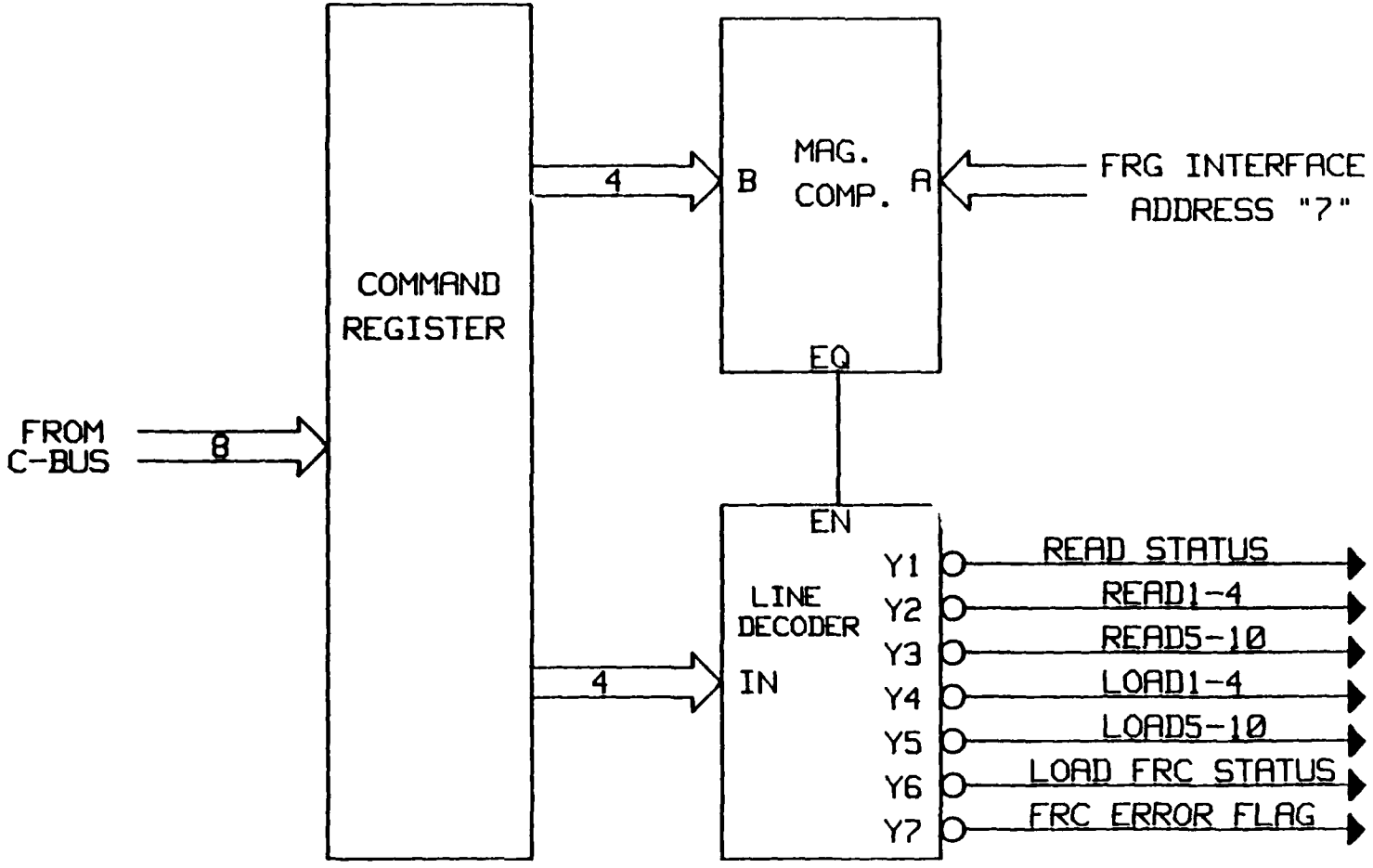


FIGURE 6.4.2.4.2-1. INTERFACE CONTROL BUS DECODING LOGIC

TABLE 6.4.2.4.2-1. DEVICE ADDRESS AND OP CODE FOR THE FRG INTERFACE

COMMAND	DEVICE ADDRESS 7	6	5	4	OP CODE 3	2	1	0
READ INTERFACE STATUS	0	1	1	1	0	0	0	1
READ1-4	0	1	1	1	0	0	1	0
READ5-10	0	1	1	1	0	0	1	1
LOAD1-4	0	1	1	1	0	1	0	0
LOAD5-10	0	1	1	1	0	1	0	1
LOAD CONTROLLER STATUS	0	1	1	1	0	1	1	0
GENERATE INTERRUPT	0	1	1	1	0	1	1	1

6.4.3 Analog Circuitry.

The ARIES is required to simulate modeled target replies and fruit replies and pass them to the Mode S sensor at the RF level. This is accomplished using the reply generation analog circuitry shown in figure 6.4.3-1. This circuitry consists of both IF and RF circuitry.

6.4.3.1 MAT/FAT IF Units.

Each ARIES reply generator (MAT, FAT) contains a subcircuit referred to as the IF Unit. An identical IF Unit is used for each reply generator and is mounted on its own analog panel (see figure 5-3, Analog Chassis Layout). The view of each side of the reply generator IF Unit is shown in figures 6.4.3.1-1 and 6.4.3.1-2.

Figure 6.4.3.1-3 is a block diagram of the IF Unit. Each such unit contains a crystal-controlled 60-MHz oscillator with each oscillator being offset from the others by 500 kilohertz (kHz). The output of the oscillator is modulated by the PAM switching module consisting of two pin diode switches incorporated in a strip line circuit to obtain on/off ratios in excess of 100 dB. The output from the modulator is attenuated in a digitally controlled diode attenuator to control the reply power level. The attenuator has a 63-dB dynamic range in 1-dB steps. The attenuated signal is divided into the main and omnidirectional (OMNI) simulated antenna channels.

The main antenna channel goes to a single pole double throw (SPDT) diode switch that allows the option of simulating mainbeam or sidelobe signal levels at the sensor input. The sidelobe level can be adjusted manually with the sidelobe gain attenuator over a 15-dB dynamic range in 1-dB steps from the preset level of -35 dB below the mainbeam. The resulting signal is then power divided to form the sum (SUM) and delta (DIFF) channels feeding the combiner.

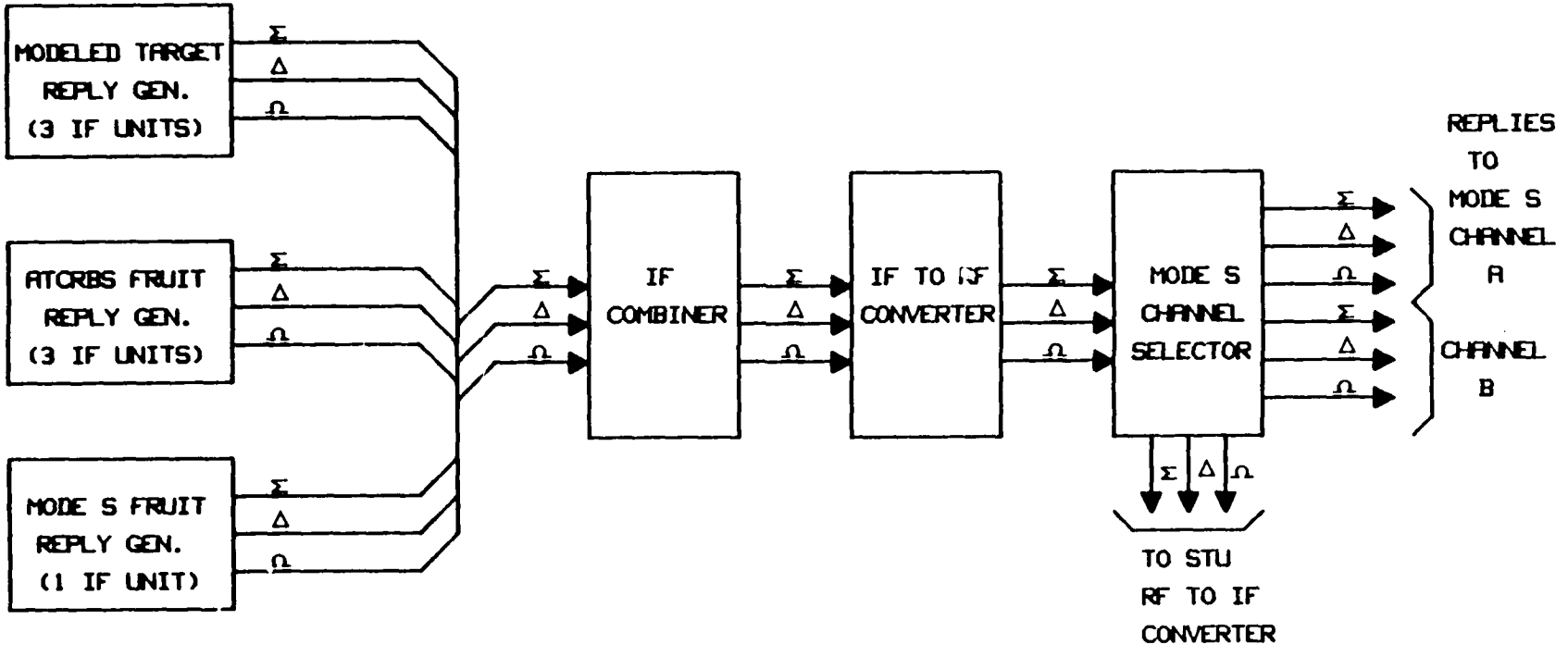


FIGURE 6.4.3-1. REPLY GENERATOR ANALOG BLOCK DIAGRAM

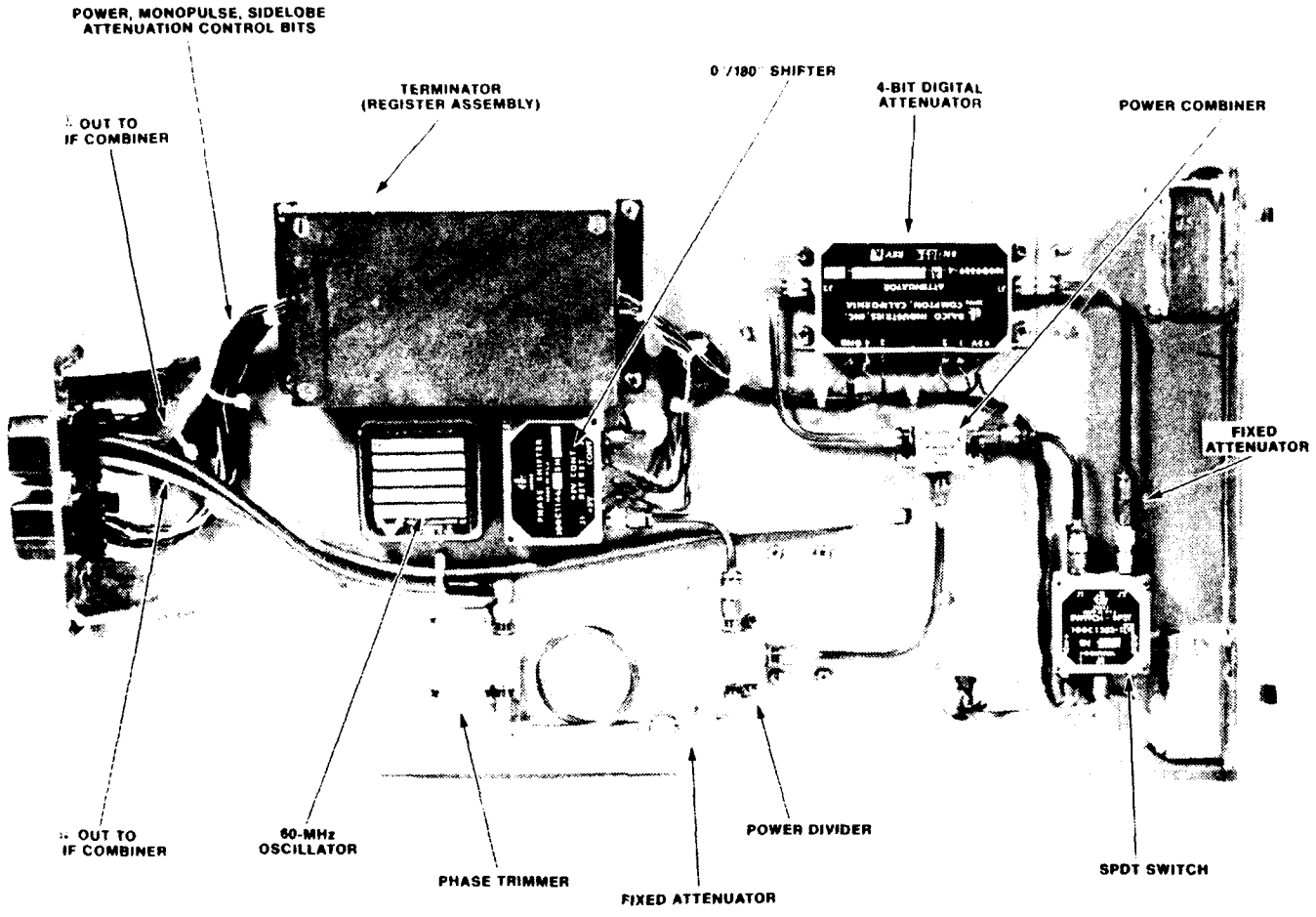


FIGURE 6.4.3.1-1. IF UNIT, SIDE A VIEW

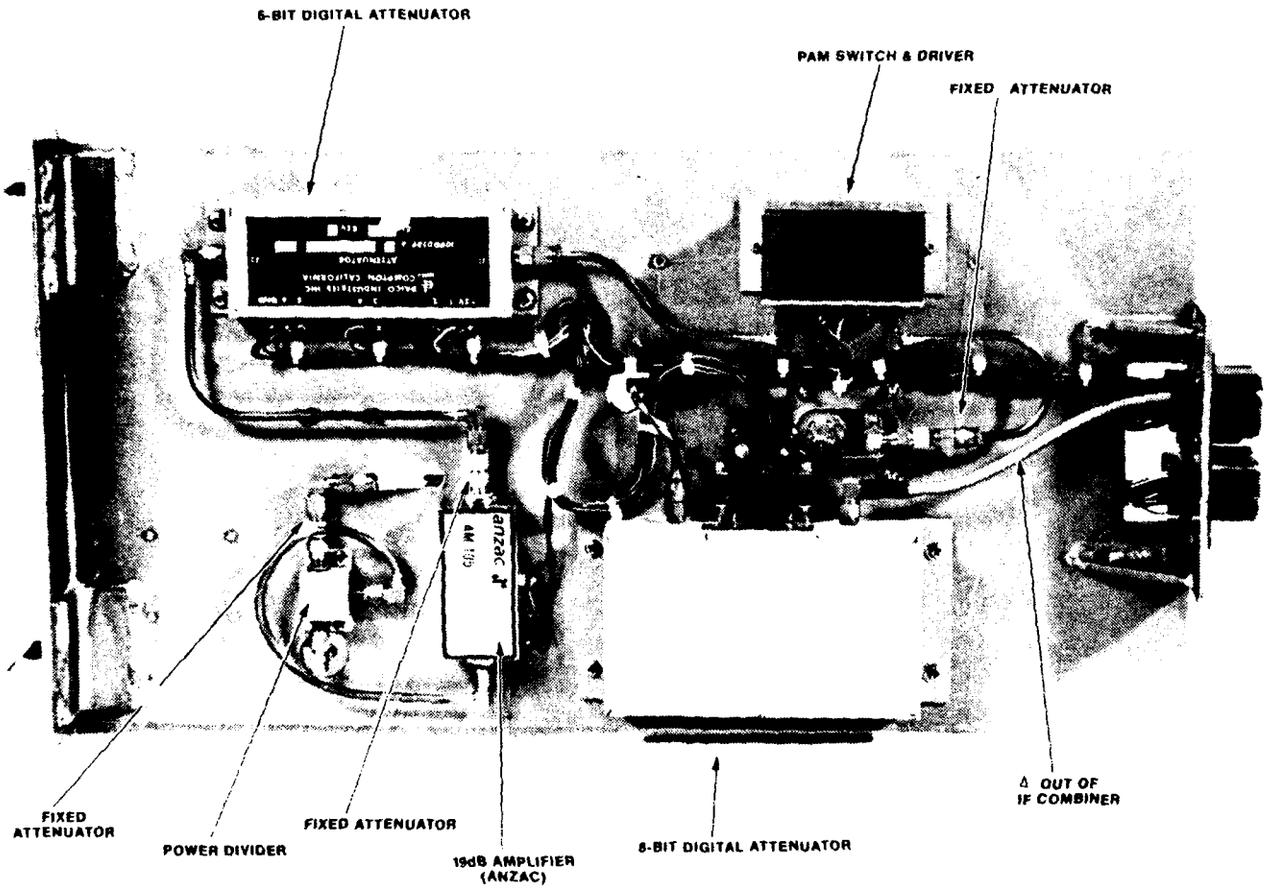


FIGURE 6.4.3.1-2. IF UNIT, SIDE B VIEW

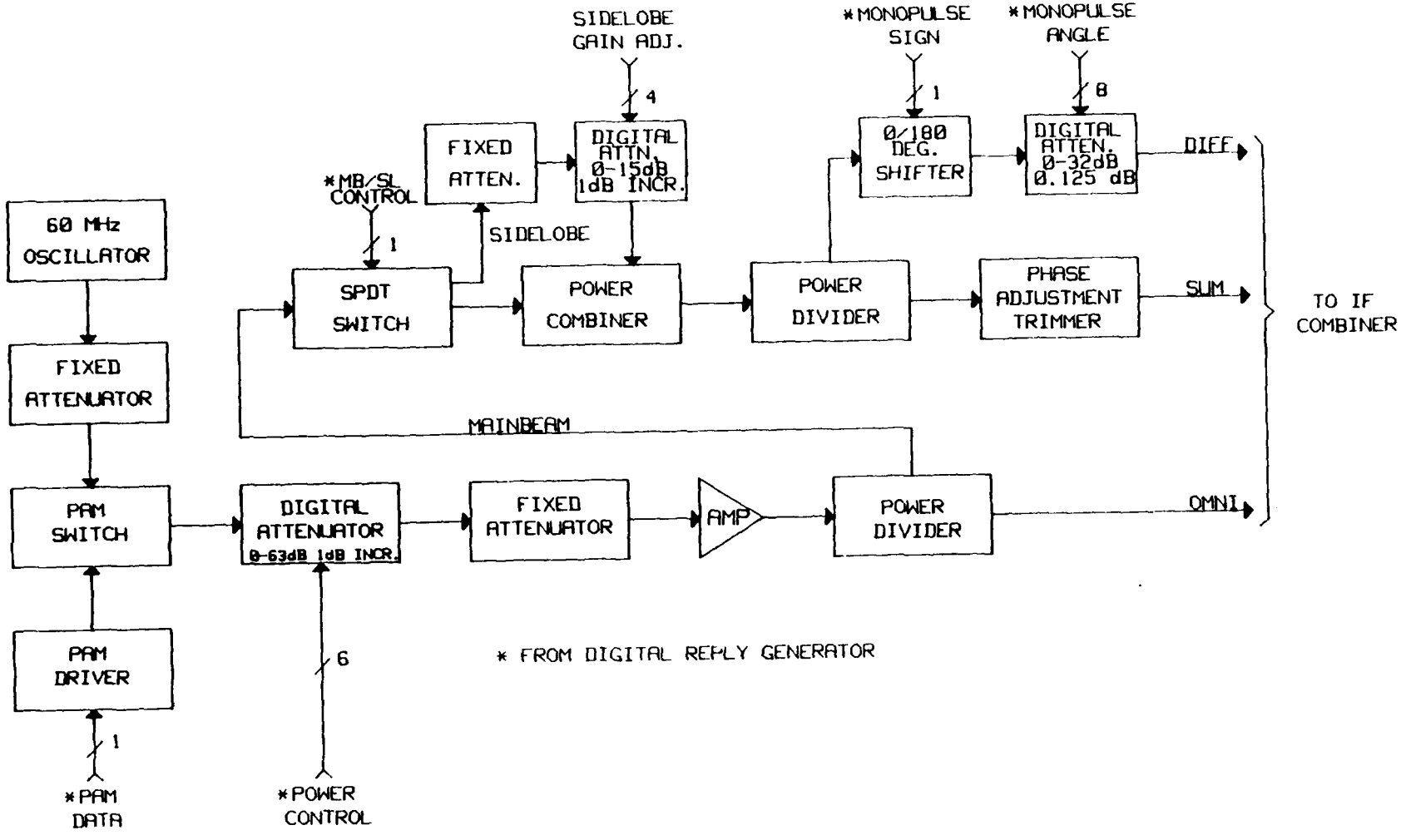


FIGURE 6.4.3.1-3. *IF UNIT BLOCK DIAGRAM*

The offboresight angle simulations are done by changing the gain setting of the DIFF channel with respect to the SUM channel. This is accomplished by means of an 8-bit digitally controlled attenuator. This attenuator has a 47.75-dB dynamic range with a resolution of 0.25 dB. At 0 attenuation, the DIFF signal is 6 dB above the SUM signal. The sign of the offboresight angle is simulated digitally by controlling a 180° phase shifter (L/R = 0 and L/R = 1 correspond to left and right of boresight, respectively). A fixed attenuator is installed in the SUM channel to compensate for losses contributed by the phase shift network in the DIFF channel, allowing amplitude balance of the SUM and DIFF channels. An adjustable phase trimmer is installed in the SUM channel to compensate for any phase differences between the SUM and the DELTA channels.

6.4.3.2 IF Combiner.

The IF Combiner, figure 6.4.3.2-1, combines separately the SUM, DIFF, and OMNI outputs from the three MAT IF Units and the four FAT IF Units. A set of 8:1 power combiners provide a single set of IF outputs for the remaining analog circuitry on the IF Combiner. Each 8:1 combiner has provisions for injecting a test signal in the eighth port for diagnostic or additional interference testing. Each power combiner output signal is attenuated by a set of manually controlled 4-bit digital attenuators. These attenuators are used to establish the relative mainbeam to omni levels for the particular Mode S antenna being used. The attenuators provide a dynamic range of 15 dB in 1-dB steps. The SUM and DIFF signals are amplified by a 28-dB linear gain amplifier. The SUM, DIFF, and OMNI signals are coupled to the front panel for monitoring, the STU analog receiver, and to the IF to RF Converter, covered in the next section.

6.4.3.3 IF to RF Converter.

The IF to RF Converter, figure 6.4.3.3-1, converts the 60-MHz IF SUM, DIFF, and OMNI signals from the IF Combiner to 1090-MHz RF output signals. This subcircuitry is mounted on its own panel (see figure 5.3, analog panel No. 12). The output from a single 1030 MHz LO is power divided into three coherent outputs, which in turn drives the three double-balanced LO mixer inputs. Each power divider output is filtered and passed through separate isolators before being applied to the mixer LO input. Each double-balanced mixer modulates the 60-MHz signals to 1090-MHz RF outputs. The outputs are passed through separate isolators and amplifiers before being coupled to the Downlink Channel Selector panel.

6.4.3.4 Downlink Channel Selector.

The Downlink Channel Selector is mounted on its own panel (see figure 5.3, analog panel No. 10). The circuitry is shown in figure 6.4.3.4-1. The SUM, DIFF, and OMNI RF signals are received from the IF to RF Converter and passed through 1090-MHz band pass filters, then coupled to the front panel for monitoring, to the STU RF to IF Convertor for signal verification, and to SPDT RF switches. These switches are controlled by the digital channel selection logic resident on the Uplink Receiver board. (See section 6.2.2.1 for details on the Channel Selection logic.) The purpose for the RF switches is to direct output replies to the active receiver of a Mode S sensor connected to a back-to-back antenna system. The active receiver is the receiver associated with the transmitter that last generated an interrogation. Each set of outputs from the RF switches are cabled to the Mode S distribution panel. An RF phase trimmer is installed on each SUM channel after the RF switch to compensate for small differences between the SUM and DIFF cable lengths connected between the ARIES and the Mode S sensor.

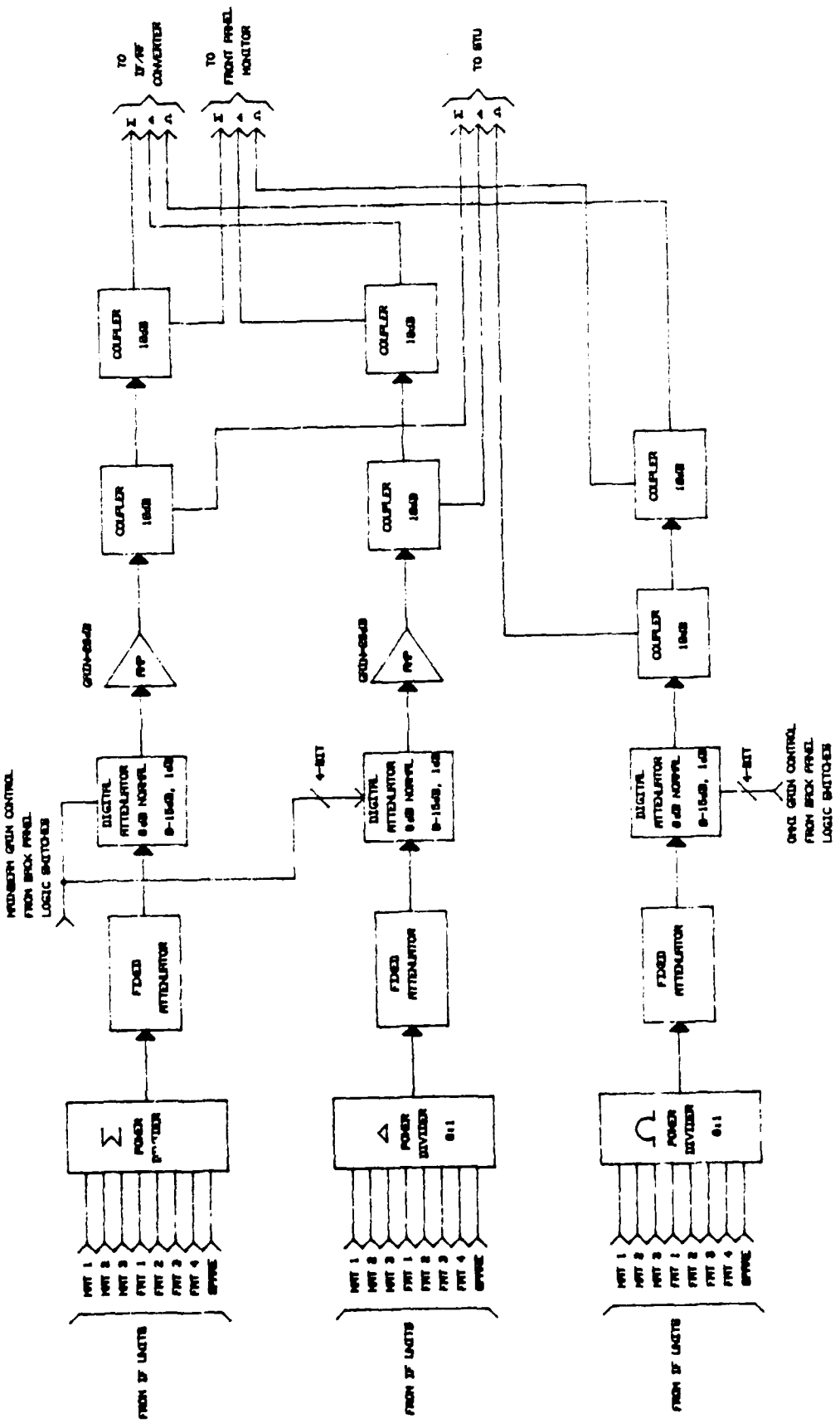


FIGURE 6.4.3.2-1. IF COMBINER BLOCK DIAGRAM

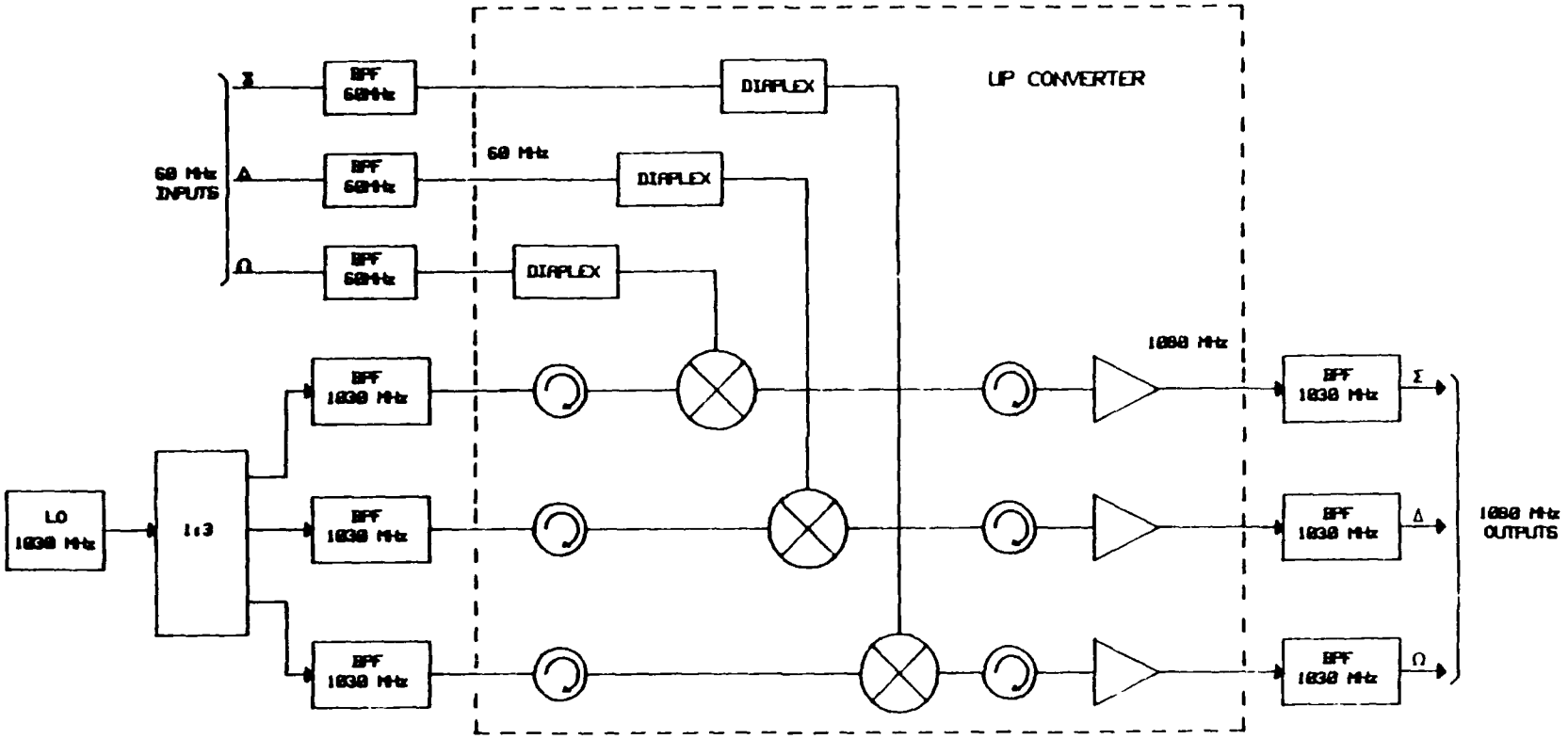


FIGURE 6.4.3.3-1. IF TO RF CONVERTER BLOCK DIAGRAM

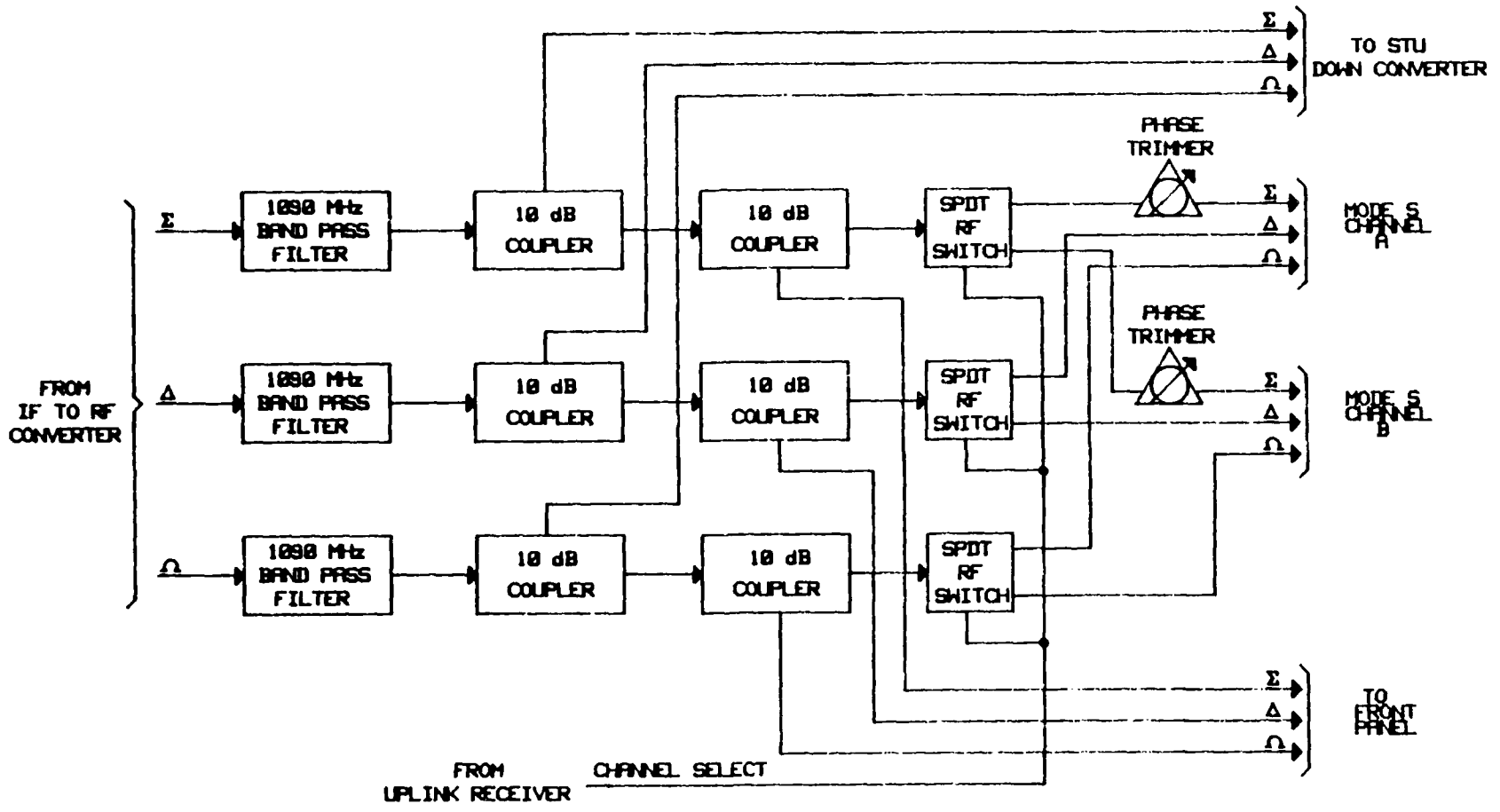


FIGURE 6.4.3.4-1. DOWNLINK CHANNEL SELECTOR BLOCK DIAGRAM

6.5 UNIVERSAL INTERVAL TIMER.

As the name suggests, the UIT is basically a timer. In its normal operating mode, the UIT provides pulses timed with respect to ATCRBS type interrogations to the Uplink Receiver, the MRG, and the computer. However, during ARIES validation checkout, trouble shooting, etc., the timer is used to provide precise time intervals related to other reference pulses; hence the title "Universal."

The UIT, as shown in figure 6.5-1, consists of the following:

a. 16-bit counter clocked at a 1-MHz rate and, therefore, capable of counting from 1 o 65,535 μs

b. 16-bit input register which holds the next preset value to be loaded into the counter

c. 16-bit two input multiplexer with output register to permit sampling the contents of the input register or the counter without disturbing its operation

d. Control logic

The UIT communicates with the computer over the standard Concurrent GPI discussed in the beginning of section 6. It receives timing input from the 16-MHz system clock and interrupt strobes from the Uplink Receiver. (See section 6.2 and Missing Interrogation Timer, section 6.6.)

Normal operation is controlled via the command halfword. Table 6.5-1 shows the action of each of the command bits as follows:

a. Assume that the computer has already calculated and stored the pulse repetition interval (PRI) of the ATCRBS All-Call interrogations from the previous set of transmissions of the Mode S sensor, and that this interval is designated T_n. (T_n need not be equal for each transmission.)

b. The computer then uses the value T_n to calculate an initial or "count from" counter setting of;

$$(65,535 - T_n + 56)$$

where it transmits to the 16-bit input register over the standard 16-bit multiplexer bus.

c. The STOP-ON-OVERFLOW, RUN, and LOAD-ON-INTERRUPT are enabled (to logical 1) via the command.

d. Receipt of an interrupt pulse form the receiver causes the counter to load in the "count from" value stored in the input register and begin counting from this preset value.

e. When the counter reaches its maximum count (counter overflows), the counter stops counting (STOP-ON-OVERFLOW gate is enabled) and the 56 EARLY pulse is sent to the three MAT reply generators, the MRC, the MIT, the Uplink Receiver, and the computer. The 56 EARLY pulse sent to the computer is referred to as the "UIT Interrupt" pulse.

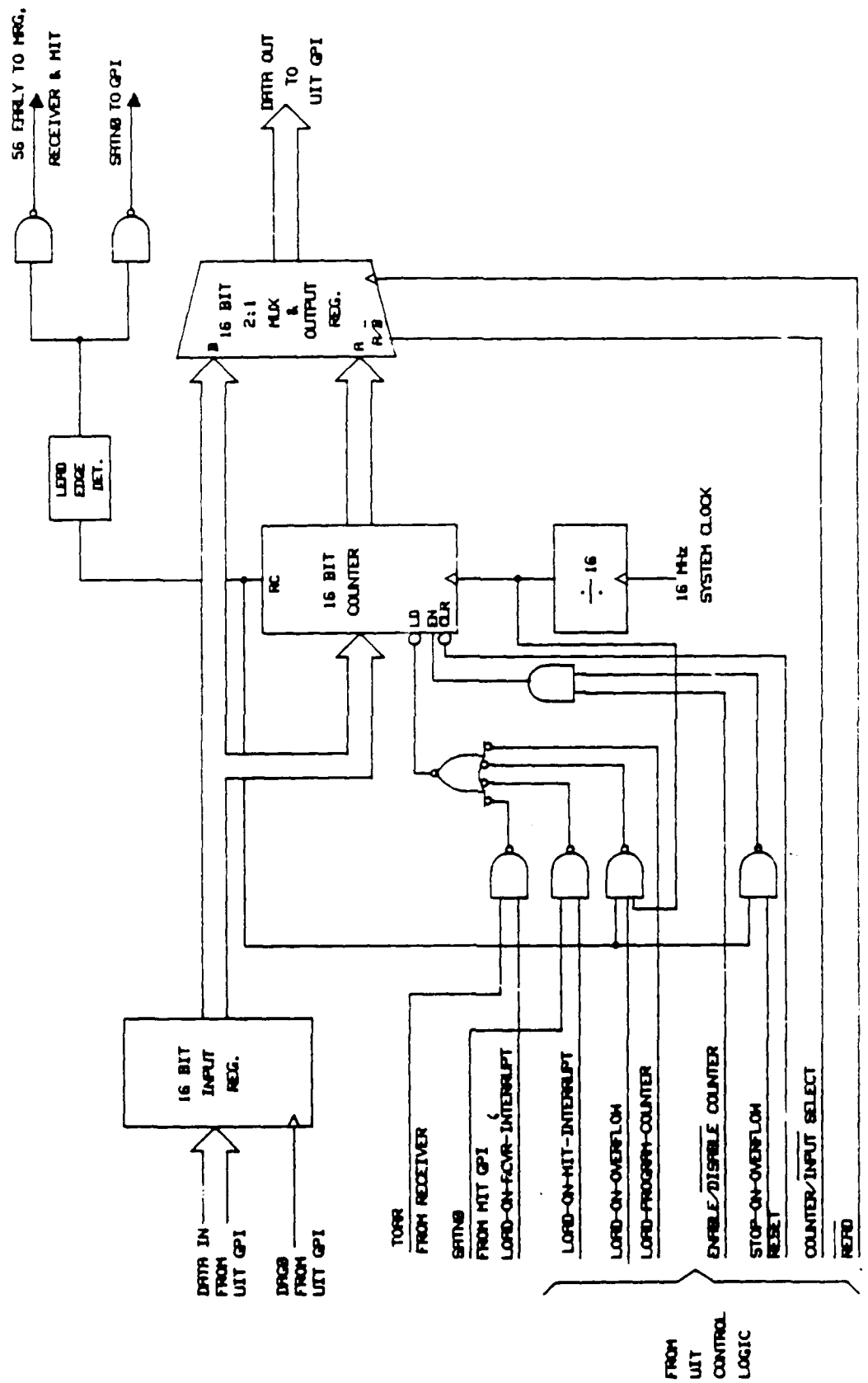


FIGURE 6.5-1. UIT BLOCK DIAGRAM

TABLE 6.5-1. UIT COMMAND BIT OPERATIONS

Bit	Command Bit Title	Action Bit Causes
7	LOAD-ON-INTERRUPT$_{(MIT)}$	When bit 10 is 0, the preset value is transferred from the input register to the counter when a MIT interrupt arrives.
10	STREEING BIT	Controls the functions of bits 7, 11, 12, and 13.
11	LOAD-ON-OVERFLOW	When bit 10 is 0, the 16-bit preset value transferred from the input register to the counter when counter overflow occurs.
	LOAD-PROGRAM-COUNTER	When bit 10 is 1, the preset value is transferred from the input register to the counter via program control.
12	LOAD-ON-INTERRUPT$_{(RCVR)}$	When bit 10 is 0, the preset value is transferred from the input register to the counter when an ATCRBS type interrogation arrives.
	$\overline{\text{RESET}}$	When bit 10 is 1, the counter is cleared to 0.
13	STOP-ON-OVERFLOW	When bit 10 is 0, the counter is stopped when overflow occurs.
	$\overline{\text{READ}}$	When bit 10 is 1, the input register or the counter selected by bit 14 is loaded into the output register.
14	CNTR/$\overline{\text{INPUT}}$ SELECT	When set to 0, the input register is to be sampled, when set to 1, the counter is to be sampled.
15	ENABLE/$\overline{\text{DISABLE}}$ CNTR	When set to 0, the counter is unconditionally frozen, when set to 1, the counter is conditionally enabled.

f. Steps "a" through "e" are repeated.

In the normal operation, reading the contents of the counter or the input register is not necessary. However, for diagnostic purposes, either may be read by setting command bits 10 and 13 to 1 and selecting the data source by bit 14. This operation will transfer the selected data into the 16-bit output register to be read by the computer via a data request.

It is sometimes desirable to have the UIT generating interrupts at constant intervals. This can be done by setting the LOAD-ON-OVERFLOW and the ENABLE COUNTER conditions (Command bit 10 set to 0, and bits 11 and 15 set to 1) and loading the value (65,535 - T) into the input register. To initialize this operation, the ENABLE COUNTER is set and LOAD-PROGRAM-COUNTER is pulsed. Therefore, the UIT will generate interrupts at a constant interval, T.

Caution must be taken not to set the UIT in an erroneous mode, i.e., setting both the LOAD-ON-OVERFLOW and the STOP-ON-OVERFLOW conditions.

The UIT 16-bit counter can be reset to 0 at any time by the RESET command (Command bit 10 and 12 set to 1).

6.5.1 UIT Control Circuitry.

The UIT control circuitry provides the means of controlling the various operations of the UIT described in the previous section. It consists of two registers and three signal-to-pulse generators shown in figure 6.5.1-1. Register A is used to hold the unconditional modes of the UIT; ENABLE/DISABLE COUNTER and COUNTER/INPUT SELECT. These modes are changeable each time the command instruction is received. Register B holds all of the conditional modes of the UIT; LOAD-ON-RCVR-INTERRUPT, LOAD-ON-MIT-INTERRUPT, LOAD-ON-OVERFLOW, and STOP-ON-OVERFLOW. These modes of operation are changed only if steering bit 10 is set low when the command is received.

The signal-to-pulse generators generate the following pulses; RESET, READ, and LOAD-PROGRAM-COUNTER. These pulses are generated if steering bit 10 is set high when the command is received. Each generator automatically resets after the pulse is generated eliminating the need to be reset between consecutive pulse generations.

6.6 MISSING INTERROGATION TIMER.

This device provides the ARIES with the capability of reacting quickly to prepare for the next predetermined interrogation if an ATCRBS/Mode S All-Call interrogation is not detected for up to three missed interrogations (software parameter nominally set at 2) before attempting to reestablish the All-Call interrogation pattern. If an All-Call interrogation is not detected over an anticipated time interval, the MIT times out initiating the following actions:

a. Notifies the ARIES computer via interrupt that the anticipated All-Call interrogation was not received.

b. Sends a reset to the MRG to purge all of the replies generated for the missed All-Call interrogation.

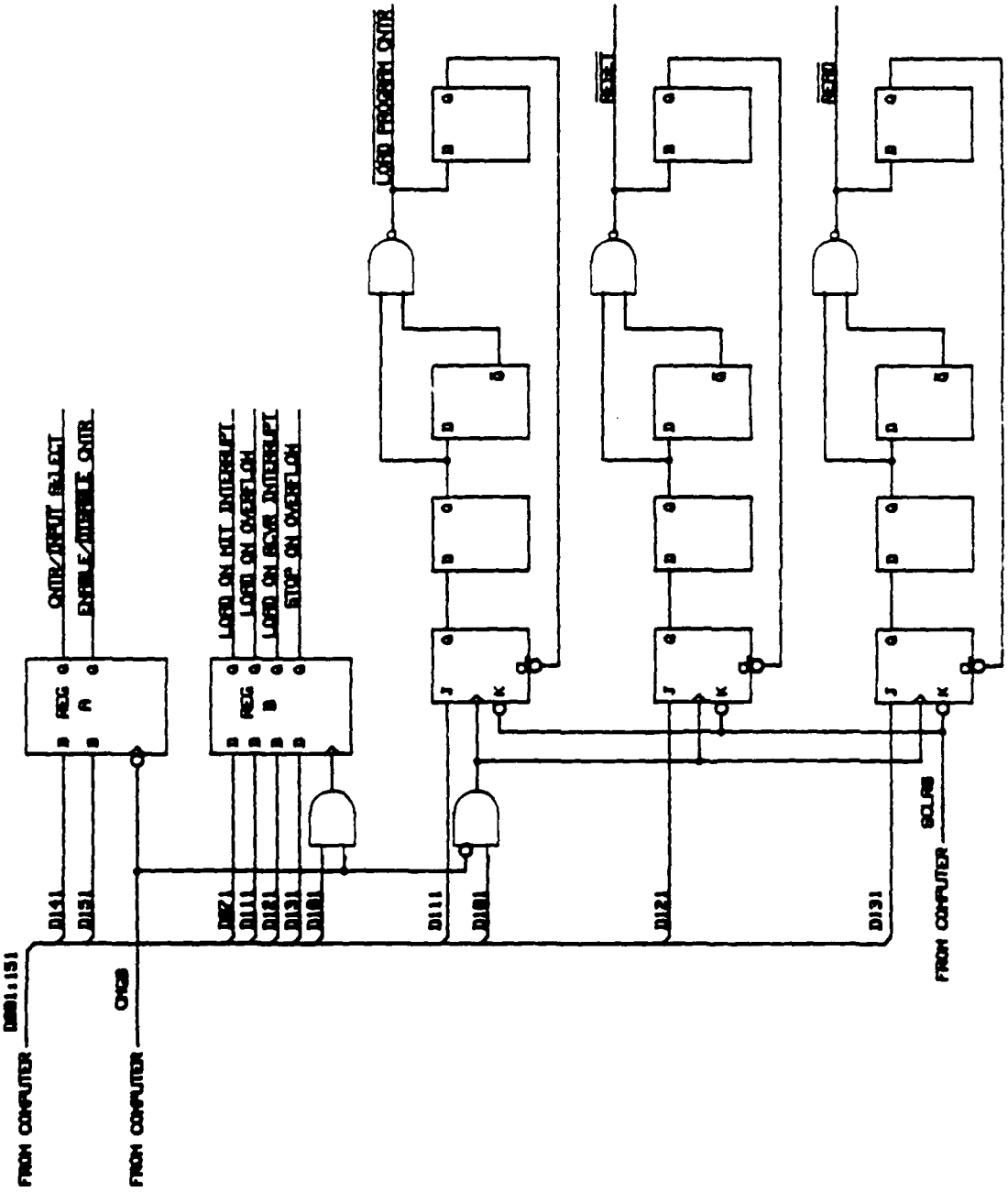


FIGURE 6.5.1-1. UIT COMMAND DECODE LOGIC

c. Triggers the UIT to start its countdown for the next scheduled All-Call interrogation.

The interrupt back to the computer is necessary not only to flag that an All-Call interrogation was missed but also initiate computer action to prepare replies for the next scheduled All-Call interrogation.

The reset to the MRG is necessary because of the way ARIES prepares ATCRBS and Mode S replies for the next scheduled All-Call interrogation, preloading them into the secondary buffer of the MRG. When the UIT generates an interrupt 56 μs before the next anticipated Uplink Receiver TOAR pulse, the MRG's current secondary buffer becomes the primary buffer and vice versa. At this time, the MRC begins transferring replies to its three MATs until they are full or until no additional replies remain in the primary buffer. If the TOAR pulse is not generated, the replies loaded in the MATs will be incorrectly transmitted for the next interrogation.

As an example, assume that the scheduled interrogation is a Mode A All-Call interrogation and that the appropriate replies were prepared and loaded into the MRG when the UIT 56 μs early pulse occurred. No Mode A All-Call interrogation was detected, so no replies are transmitted. Now assume that the interrogation mode pattern is A-C-A-C, so the next scheduled interrogation would be a Mode C All-Call interrogation. When this interrogation is detected, the replies loaded into the MATs, prior to the Mode A All-Call, would be transmitted even though the actual interrogation detected was a Mode C All-Call. Therefore, the reset from the MIT is used to purge all of the preloaded replies generated for the missed interrogation.

The trigger to the UIT is necessary to substitute for the nongeneration of the trigger pulse from the Uplink Receiver to start the UIT for the next countdown for the next scheduled interrogation.

The MIT accomplished this by being programmed to timeout 10 μs after the scheduled detection of the All-Call interrogation. Like a simplified version of the UIT, the MIT is a precision timer with interrupt capability. In the normal mode of operation, the MIT is started by receiving the 56 EARLY pulse from the UIT. It increments until it reaches its maximum value at which time a 1-clock wide timeout pulse is generated. When the scheduled All-Call interrogation is detected by the Uplink Receiver, a TOAR pulse is generated and fed back to the MIT preventing the MIT timeout pulse from being issued. If the TOAR pulse is not generated, the MIT remains enabled and the timeout pulse is allowed to be issued. The timing action of this sequence is illustrated in figure 6.6-1.

The MIT logic as shown in figure 6.6-2. consists of the following:

a. 8-bit binary counter clocked at 1-MHz rate and therefore capable of counting from 1 to 256 μs

b. 8-bit input register which holds the preset value to be loaded into the counter

c. Control circuitry

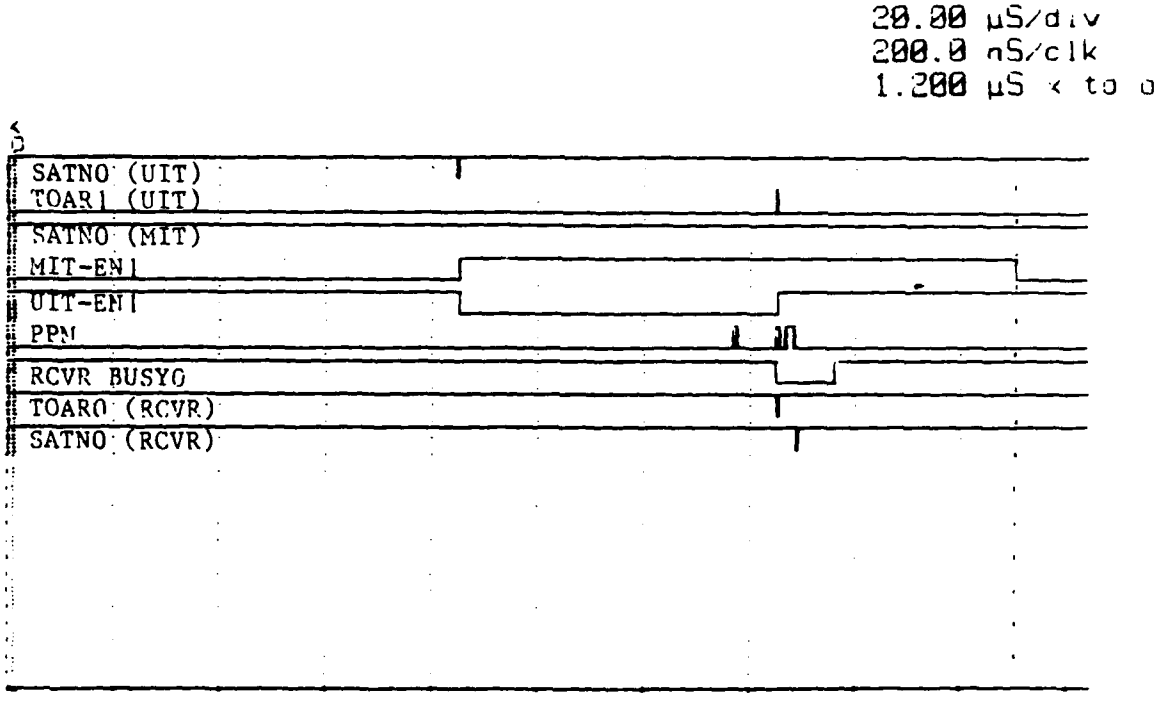


FIGURE 6.6-1. INTERACTIVE TIMING WAVEFORMS BETWEEN RCVR, UIT AND MIT

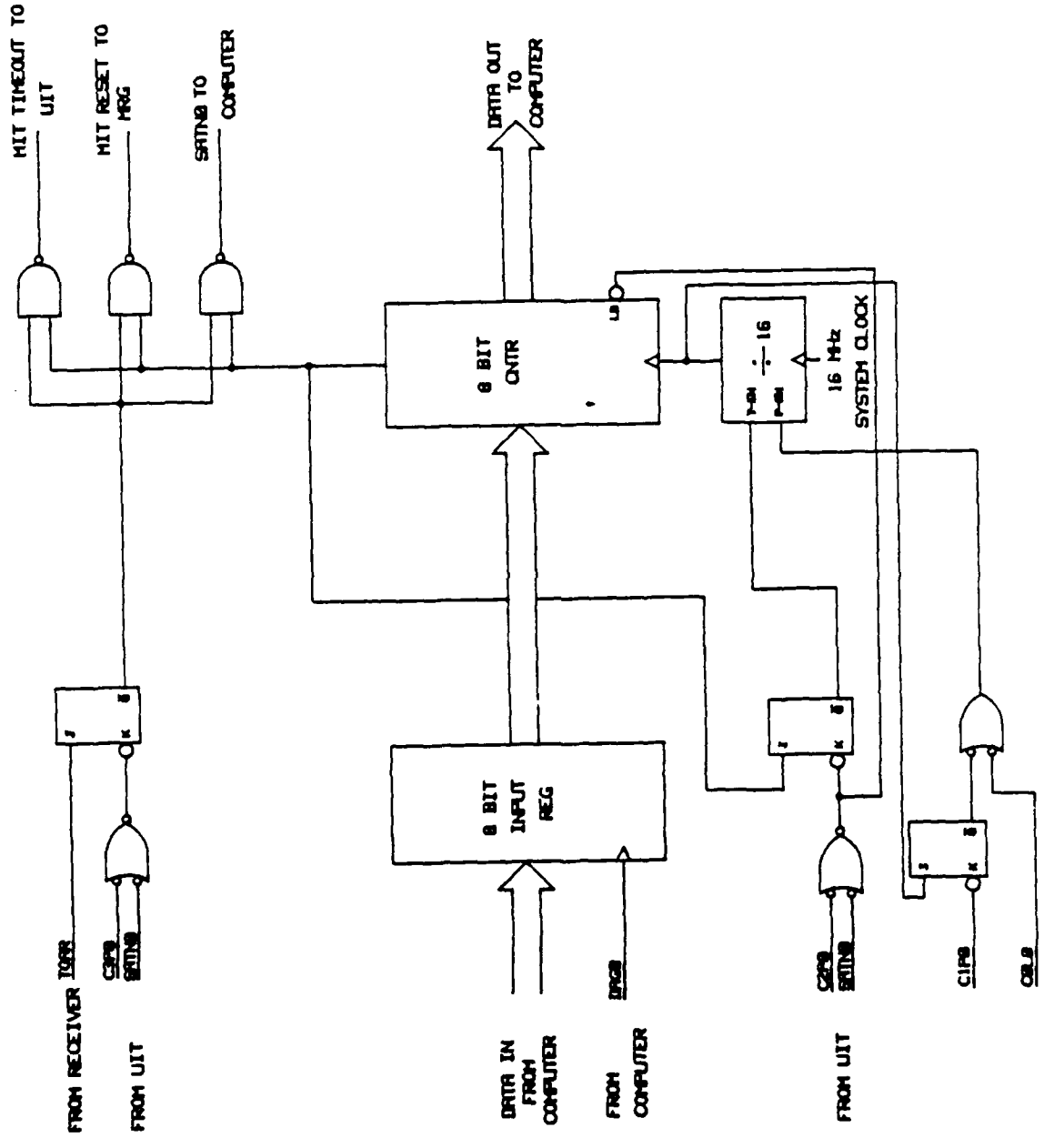


FIGURE 6.6-2. MIT BLOCK DIAGRAM

6.6.1 MIT Control Circuitry.

The MIT control circuitry provides the means of controlling the various operations of the MIT. It consists of three signal-to-pulse generators which produce control pulses C1P0, C2P0, and C3P0 and a J/K flip-flop used as a 1-bit register shown in figure 6.6.1-1. This circuitry is used for validation and trouble shooting of the MIT as a stand-alone device. The C3P0 control pulse enables the MIT to transmit its timing pulses to other devices. The C2P0 control pulse loads the preset value into the MIT counter and resets the T-EN flip-flop, conditionally enabling the counter. In this mode, the C1P0 pulse is used to increment the counter by 1, or the C0L0 signal is set to allow the counter to increment at a 1-MHz clock rate. When the counter reaches its maximum count, the ripple carry is generated and is fed back to the T-EN flip-flop. On the next clock cycle, the flip-flop is set stopping the counter.

6.7 RANDOM NUMBER GENERATOR.

The RNG provides the computer with 16-bit positive uniformly distributed random integers. It consists of 5 T-type flip-flops and 12 D-type flip-flops cascaded as shown in figure 6.7-1. With output of the last D flip-flop fed back to the input to the first T flip-flop, a 17-bit maximal length sequence generator is formed.

Each time the RNG is read by the computer (DR0 instruction), it will be clocked once, and a new number is produced.

This generator is capable of generating all 2^{17}-1 possible states, except 0. Since 0 is a illegal number, causing a lockup condition, the generator must not start in this condition. To assure this, the RNG is reset to state "11100---000" during a power reset or at any time by the programmed reset instruction.

6.8 RADAR REPORT GENERATOR.

The RRG reformats radar reports resident in the computer's memory into ASR-9 or CD-2 formatted reports for transmission to the radar input port of the Mode S sensor. The format of the CD-2 output conforms to the output of a CD-2 equipped with the ASR-9 Adapter modification kit. Data transmission to the Mode S sensor conforms to the ADCCP protocol.

The complete RRG is mounted on one digital board located in slot No. 15 of the ARIES Digital Chassis. The RRG consists of the Radar Report Controller (RRC), the ADCCP Data Convertor (ADC), and the RRG Buffer Interface. A block diagram of the major modules of the RRG is presented in figure 6.8-1.

6.8.1 Radar Report Controller.

The primary function of the RRC, a high-speed bit-slice microprocessor, is to fetch report blocks from the RRG Buffer Interface, prefix the command and address fields to the blocks, and load them into the ADC for transmission.

The hardware of the RRC is identical to that of the MRC with the exception of the output circuitry. A second register was cascaded with the original output register to form a two-level output buffer. This was done to improve microcode efficiency when transferring data to the ADC. Also, output drivers were no longer necessary since the entire RRG is mounted on one digital board. The output circuitry for the RRC is shown in figure 6.8.1-1.

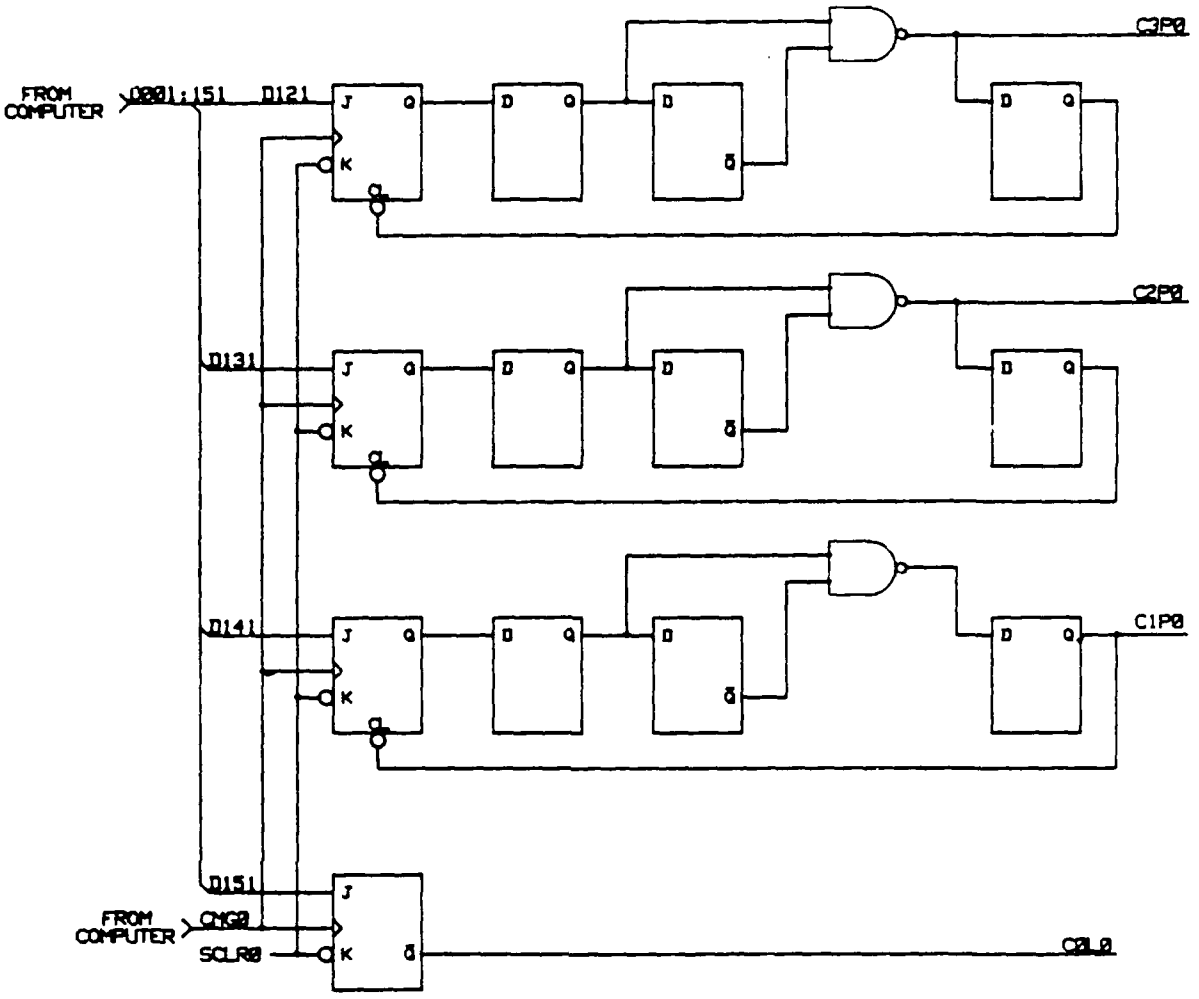


FIGURE 6.6.1-1. MIT COMMAND DECODE LOGIC

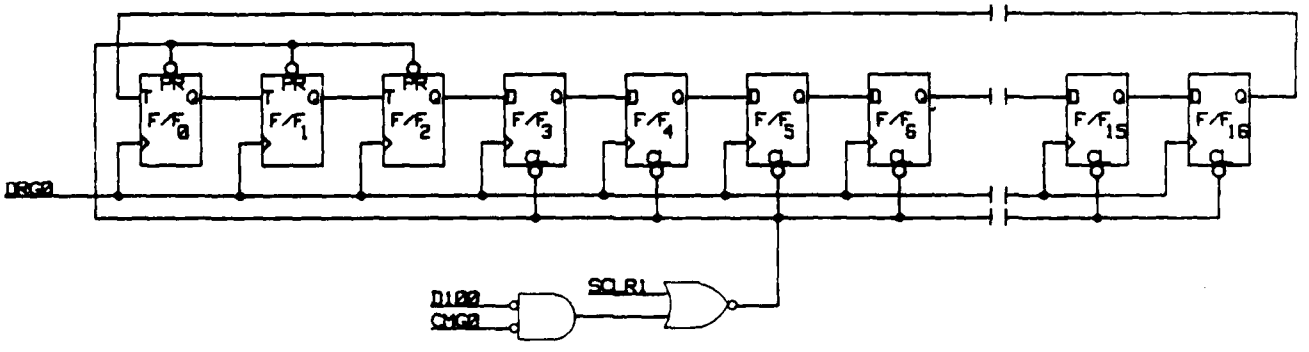


FIGURE 6.7-1. RNG BLOCK DIAGRAM

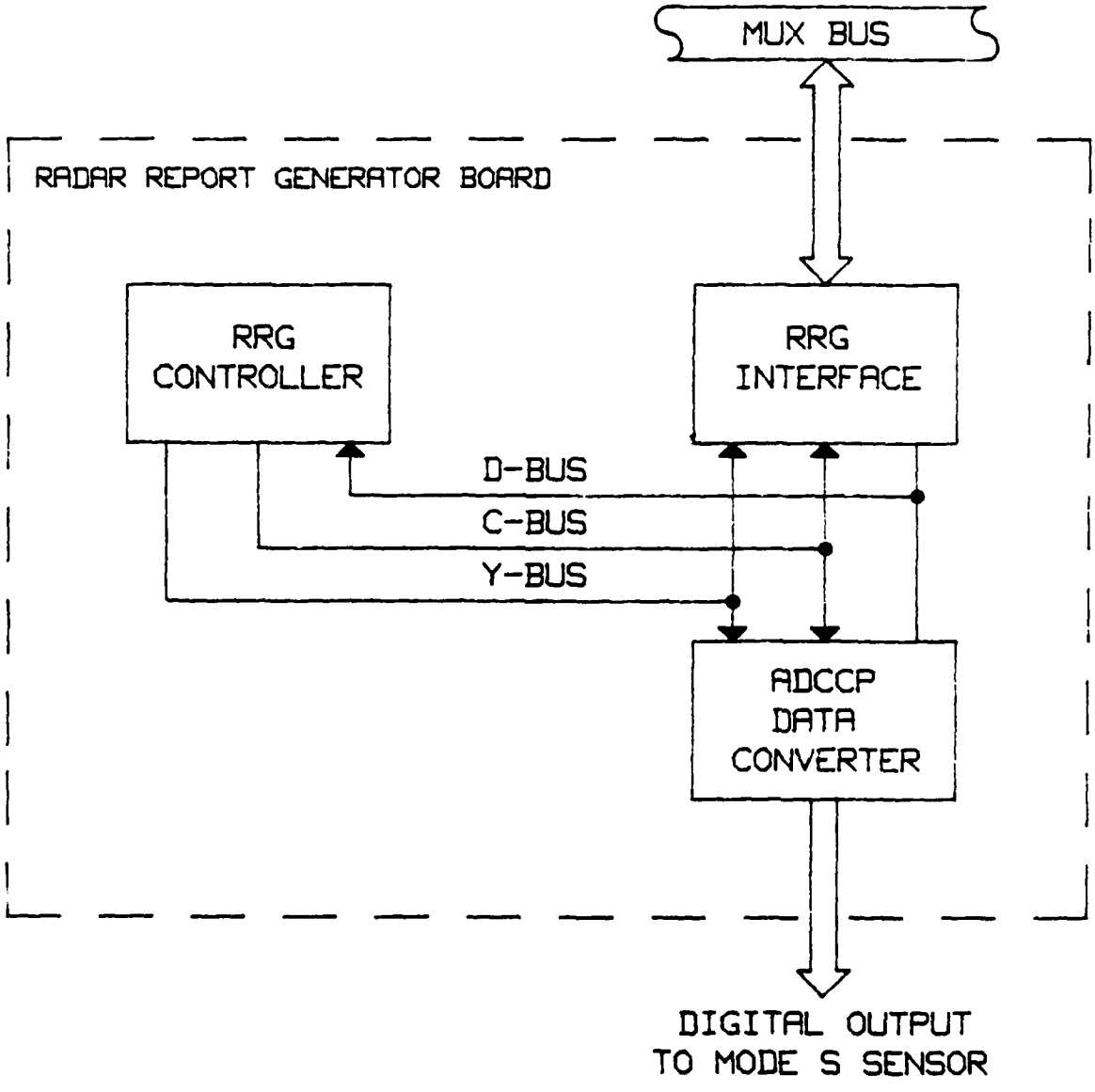


FIGURE 6.8-1. RRG DIGITAL CONFIGURATION

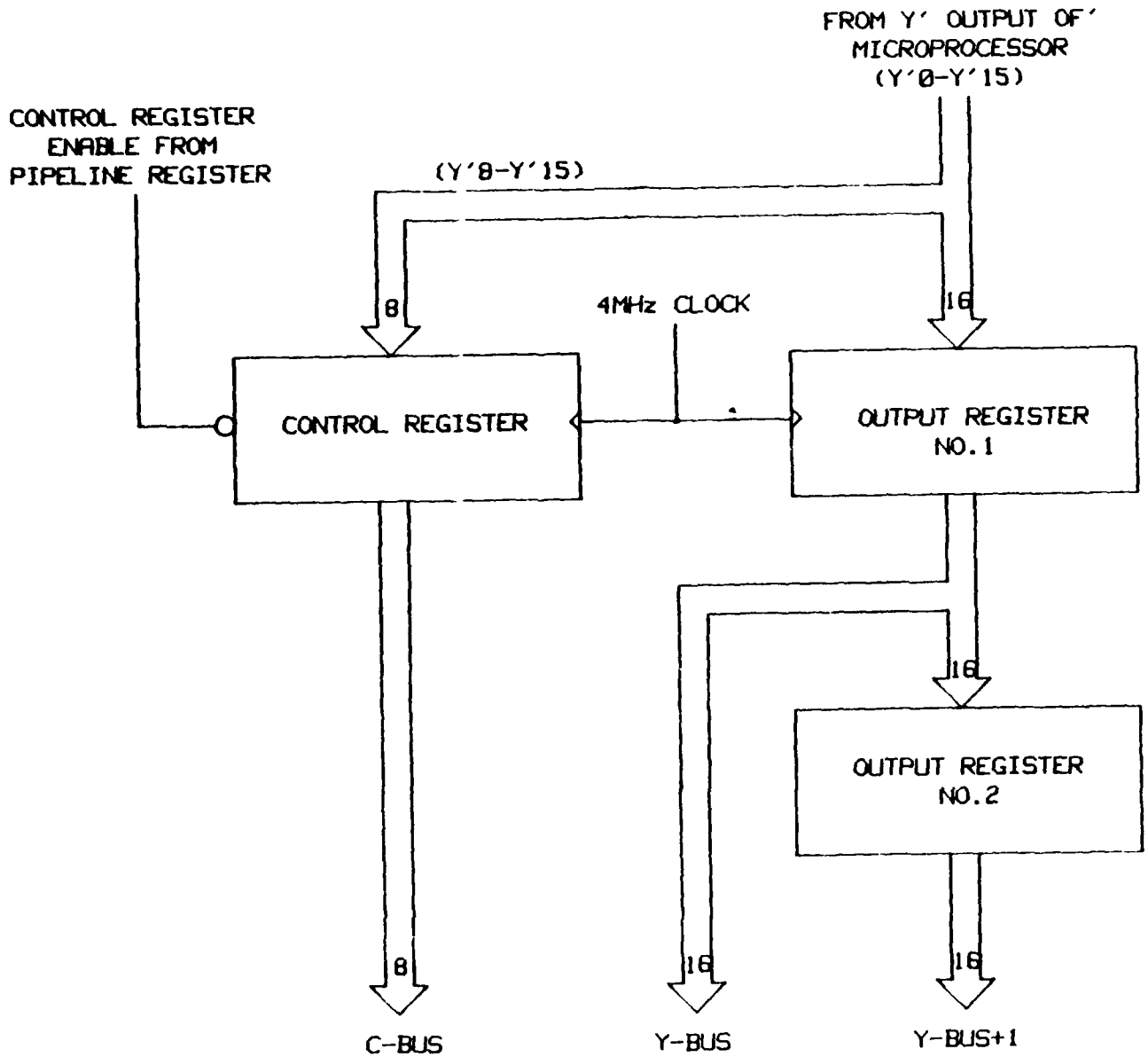


FIGURE 6.8.1-1. RRC OUTPUT CIRCUITRY

The firmware resident in the ROM does differ. (Refer to Volume II, appendix B for the RRG microprogram listings.)

6.8.2 ADCCP Data Converter.

The ADC is partitioned into transmission and diagnostic circuitry. The transmission circuitry is discussed first in section 6.8.2.1, and the diagnostic circuitry is discussed in section 6.8.2.2.

6.8.2.1 ADC Transmission Circuitry.

A functional block diagram of the ADC Transmission Circuitry is illustrated in figure 6.8.2.1-1. This circuitry transmits ASR-9 or CD-2 formatted reports to the Mode S sensor conforming to standard ADCCP protocol. The data protocol used is a subset of ADCCP as per ANSI Standard X3.66. The subset is called Asynchronous Balanced Mode which means that data transmission can occur without asking permission from the receiving system. Data is transferred synchronously at 0.5 megabits per second. No error recovery procedures are implemented. All active RS449 circuits, that is, circuits with drivers and/or receivers, are implemented using balanced differential drivers and receivers.

6.8.2.1.1 Transmission Clock Logic.

The ADC must transmit data to the Mode S Radar Port Adapter at the rate of 0.5 megabits per second. Therefore, a 500-kHz transmission clock was derived from the 4-MHz system clock employing the logic shown in figure 6.8.2.1.1-1. The counter is designed to repeat its cycle every eight system clocks. The counter continuously increments until it reaches its maximum value at which time the ripple carry pulse is produced. This pulse is inverted and fed back to the load input line causing the preset value of 8 to be loaded into the counter on the following clock cycle to repeat the sequence.

The Q_C output line is used to produce the transmission clock with a 50-percent duty cycle. This line is fed to two drivers for distribution in the ADC. Note that the output of one driver is continuous and that the output of the other driver can be interrupted by the control signal SKIP0. This clock is referred to the controlled clock (CNTL-CLK) and is used to pause the data streams when a 0 bit is inserted on the transmission line.

6.8.2.1.2 Input Buffer and Input Buffer Address Logic.

The Input Buffer logic consists of a 16-bit x 16-word memory, a 16-bit shift register, write and read address counters, a 2:1 multiplexer address selector, and an address comparator. The block diagram of the Input Buffer logic is shown in figure 6.8.2.1.2-1.

Since the Microcontroller can execute eight instructions for every data bit transmitted, stacking reports in the ADC is not necessary. The RRC has sufficient speed to detect when the ADC has completed a message transmission and load the next report into the input buffer prior to the completion of an idle frame transmission required between messages. Therefore, the input buffer need only be large enough to hold one report target, status, etc., for transmission. The input buffer is constructed of four 4-bit x 16-word deep memory ICs to handle 16-bit wide words.

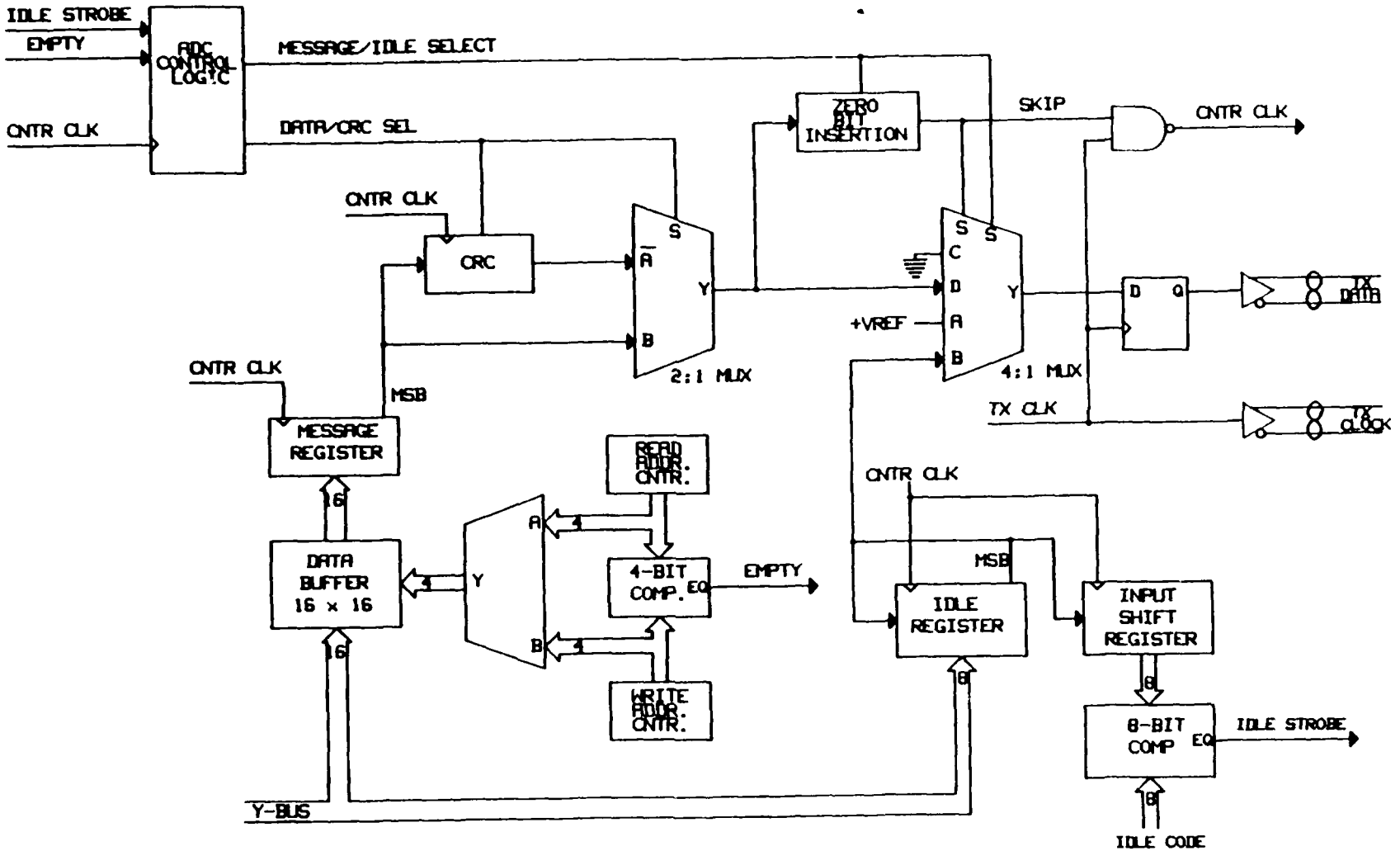


FIGURE 6.8.2.1-1. ADC BLOCK DIAGRAM

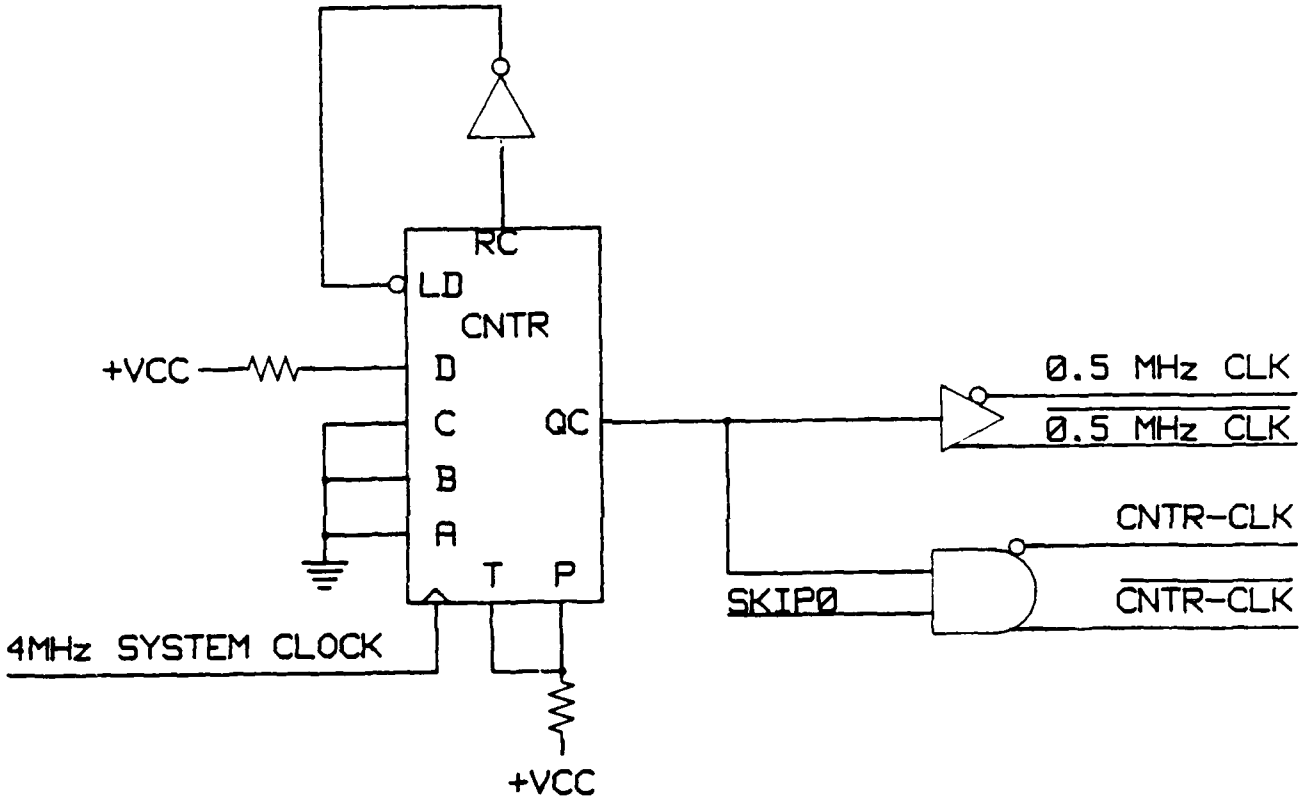


FIGURE 6.8.2.1.1-1. TRANSMISSION CLOCK LOGIC

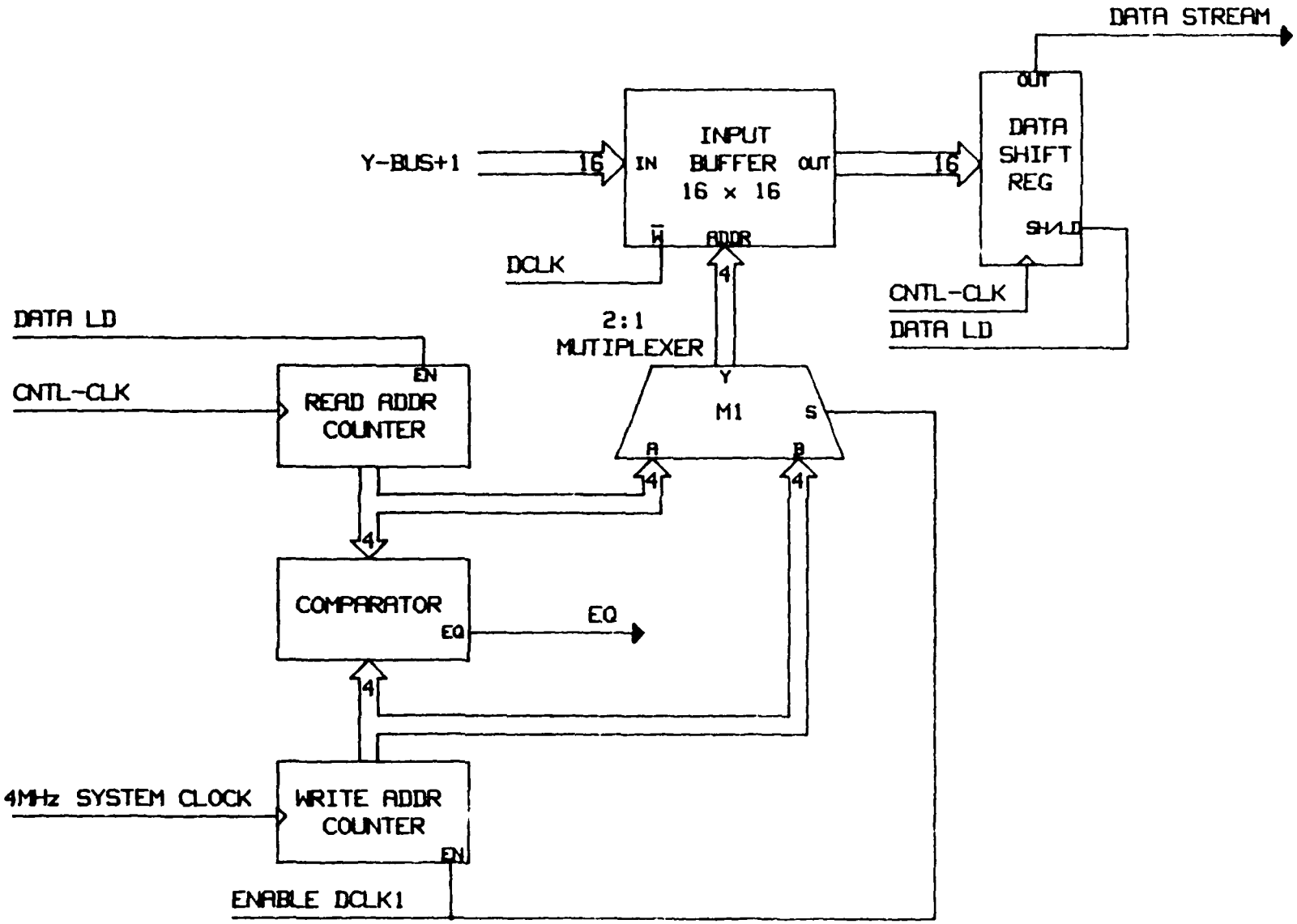


FIGURE 6.8.2.1.2-1. INPUT BUFFER BLOCK DIAGRAM

When a data load is executed, the write address is placed on the memory address lines via multiplexer M_1 and the write address counter is enabled. On the following clock cycles, a write memory strobe is produced by the RRC Command Decode logic. (See section 6.8.2.1.4.) The write address counter is incremented after each write strobe. At this time, the read/write addresses no longer match, and the output of the address comparator, test signal EQ, indicates data is present in memory. The test signal EQ is used by the ADC State Controller covered in section 6.8.2.1.7.

Once the write cycle is completed, the read address is placed back on the memory address lines. When data is transferred from the input buffer to the data shift register, the read address is stepped to the next location. This is repeated each time a complete word is shifted out of the register until all of the data is transmitted. At this time, both address counters have the identical value causing the test signal EQ to go high indicating that no additional data is available.

During system initialization, both address counters are cleared to 0. However, both counters may be set back to 0 at any time if the ADC State Controller losses synchronization with the idle frame.

6.8.2.1.3 Idle Frame Support Logic.

Referring to figure 6.8.2.1.3-1, the Idle Frame supporting logic consists of an 8-bit parallel register (REG_1), a parallel-to-serial shift register (REG_2), a serial-to-parallel shift register (REG_3), and an 8-bit magnitude comparator (COMP). In normal operations, the idle frame is loaded into REG_1 before being transferred to REG_2. This is necessary because the idle frame is transferred over the Y-Bus synchronized to the 4-MHz system clock. REG_1 holds the idle frame for transfer to REG_2 synchronized to the 0.5-MHz controlled transmission clock (CNTL-CLK).

The output line Q_H of REG_2 is used as the idle stream output and is fed back to the serial input line setting up a continuous loop of idle frames through REG_2. The idle stream is taken by REG_3 and converted back into a parallel byte and placed on the P-inputs of COMP. The Q-inputs of COMP are strapped to match the idle frame. The strapping is shifted by two clock cycles to compensate for a two-clock delay in the idle stream before being transmitted. In this way, the idle strobe is generated at the start of each idle frame at the start of its transmission. The idle strobe is used as a timing alignment pulse by the ADC State Controller to correctly inject messages between idle frames.

6.8.2.1.4 RRC Command Decode Logic.

The ADC decodes four commands from the RRC. These commands are listed in table 6.8.2.1.4-1. The commands are received over the C-Bus.

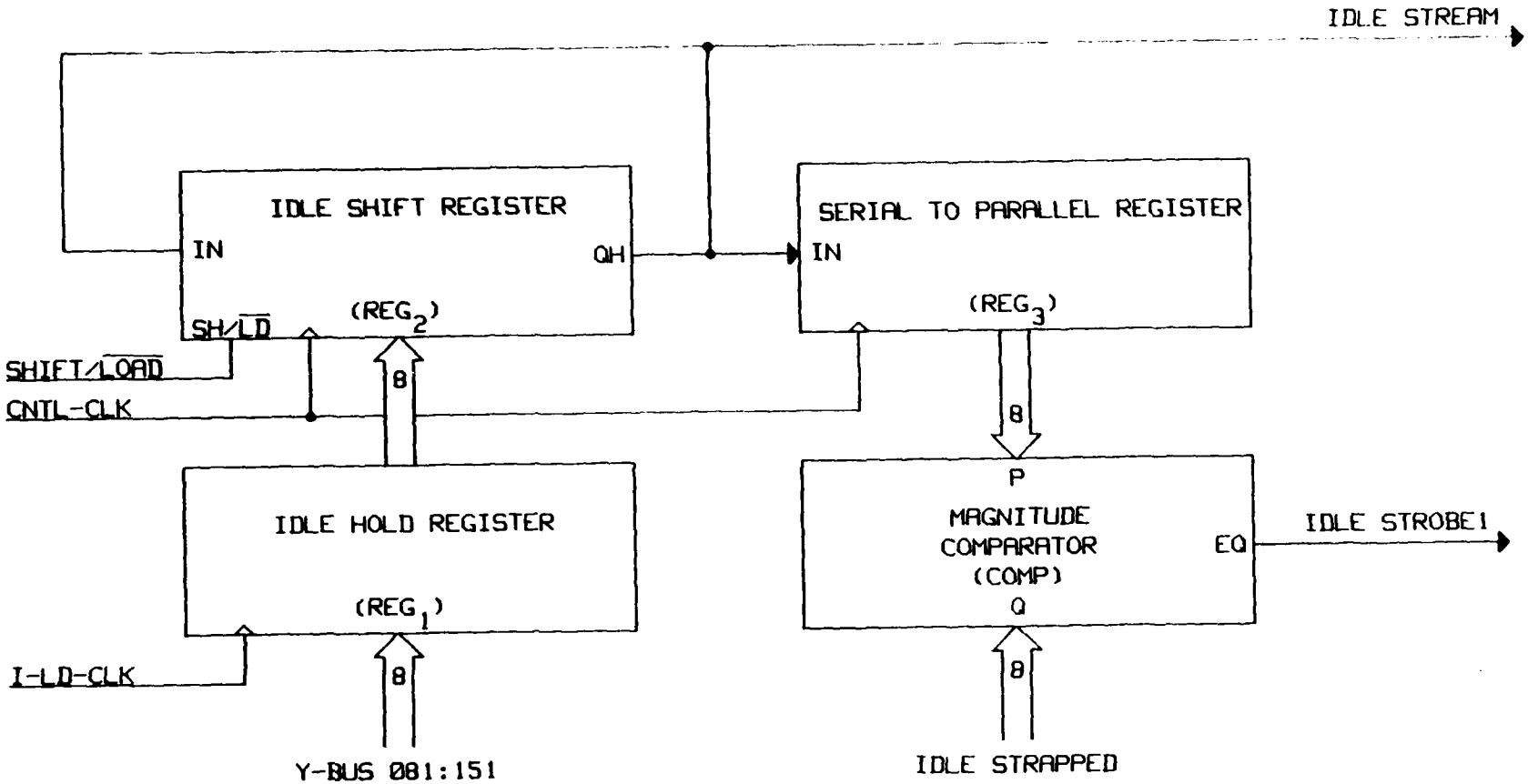


FIGURE 6.8.2.1.3-1. IDLE FRAME SUPPORT LOGIC BLOCK DIAGRAM

TABLE 6.8.2.1.4-1. DEVICE ADDRESS AND OP CODE FOR THE ADC

COMMAND	DEVICE ADDRESS				OP CODE			
	7	6	5	4	3	2	1	0
READ DATA	0	0	0	0	0	1	0	0
DATA LOAD COMPLETE (GO)	0	0	0	0	0	1	0	1
LOAD IDLE REGISTER	0	0	0	0	0	1	1	0
LOAD DATA	0	0	0	0	0	1	1	1

The load idle pulse (LD-IDLE0) is used to load a byte into the Idle Register and subsequently transfer it to the Idle Shift Register. This is accomplished by the decode logic shown in figure 6.8.2.1.4-1 alone with the timing waveforms. When the load idle command is decoded, it is delayed by two clock periods to align the load clock, I-LD-CLK, with the data placed on the Y-Bus. At the same interval, the LD-IDLE0 signal is fed to a J/K flip-flop IC_1 which is set on the following 4-MHz clock cycle. The Q-output of IC_1 is fed to the D flip-flop IC_2 to synchronize the LD-IDLE0 signal to the 0.5-MHz transmission clock. On the following 0.5-MHz clock cycle, IC_2 is set and control signal SH/LD goes low placing the Idle Shift Register in the load mode. The output of the Idle Register is transferred into the Idle Shift Register completing the load on the following low-to-high transition of the 0.5-MHz clock.

The load data pulse (LD-DATA0) is used to store report data in the Input Buffer. The decode logic is designed to accommodate the loading of any number of words automatically based on how long the load command remains active on the C-Bus. (Note the data transfer must be limited to 16 words or less because of the input buffer size of 16.) This is accomplished by the logic shown in figure 6.8.2.1.4-2 along with its timing waveforms. When the load data command is decoded, the LD DATA0 control signal is fed to leading edge and trailing edge detectors. The produced leading edge pulse is delayed for 1-clock cycle to align it with the incoming data on Y-Bus+1. The active high input signal is fed to the J-input of flip-flop IC_1. On the following clock cycle, IC_1 is set enabling the Input Buffer Write Address Counter and enabling NAND gate A to pass the memory write strobes (DCLK) to the Input Buffer. When the load data command is removed from the C-Bus, the LD-DATA0 control signal goes inactive high and is trailing edge detected. The trailing edge pulse is fed to the K-input of flip-flop IC_1, causing it to reset on the following clock cycle and so ending the data load sequence.

The Load Data Complete command (GO) sets a condition bit used by the ADC State Controller. This command is sent by the RRC after a complete report is loaded into the input buffer.

The read data command (READ DATA) is used in the diagnostic mode to read transmitted data as it is being fed back through built-in diagnostic hardware referred to in section 6.8.2.2.

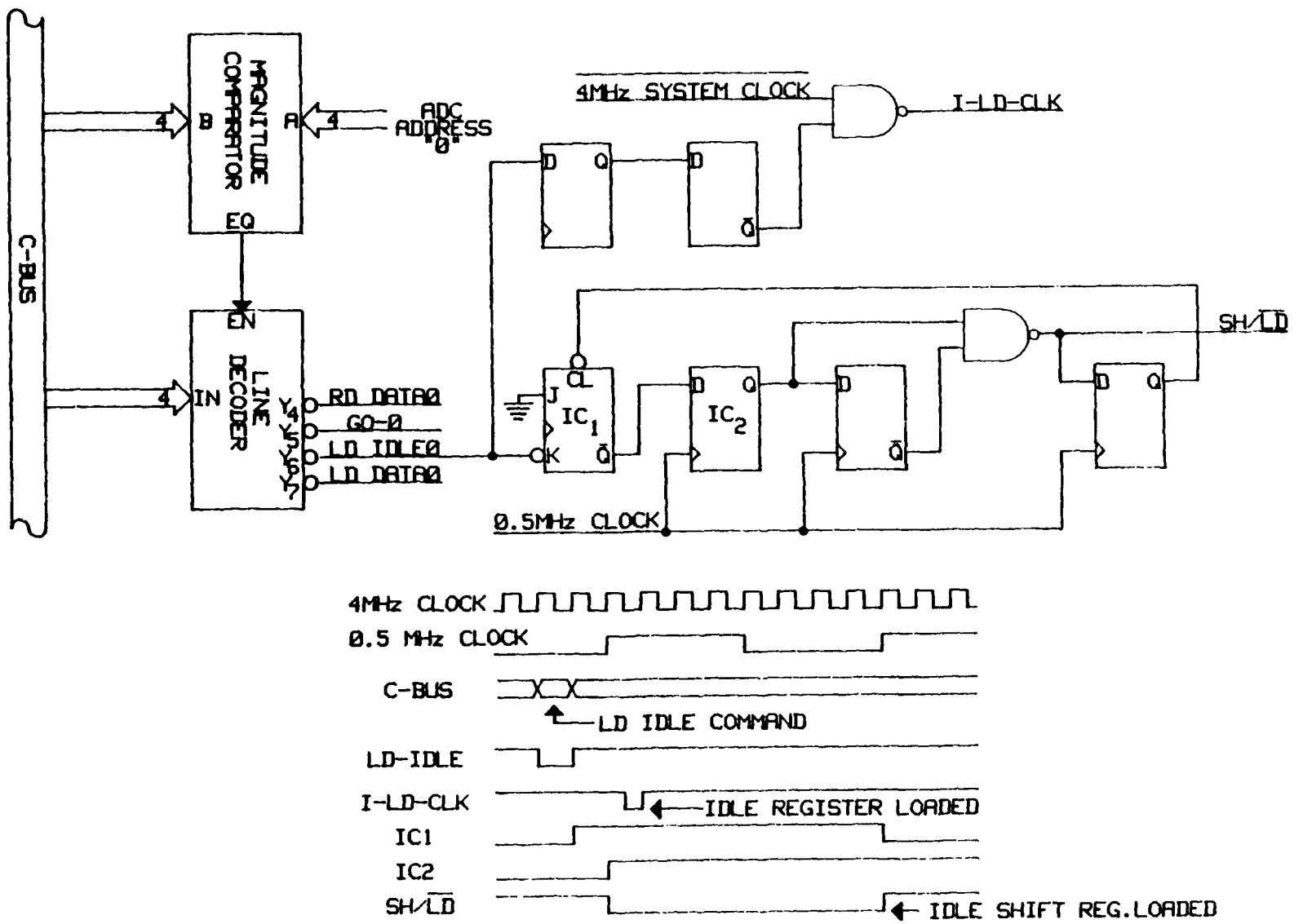


FIGURE 6.8.2.1.4-1. ADC COMMAND DECODE AND IDLE LOAD CONTROL LOGIC WITH TIMING WAVEFORMS

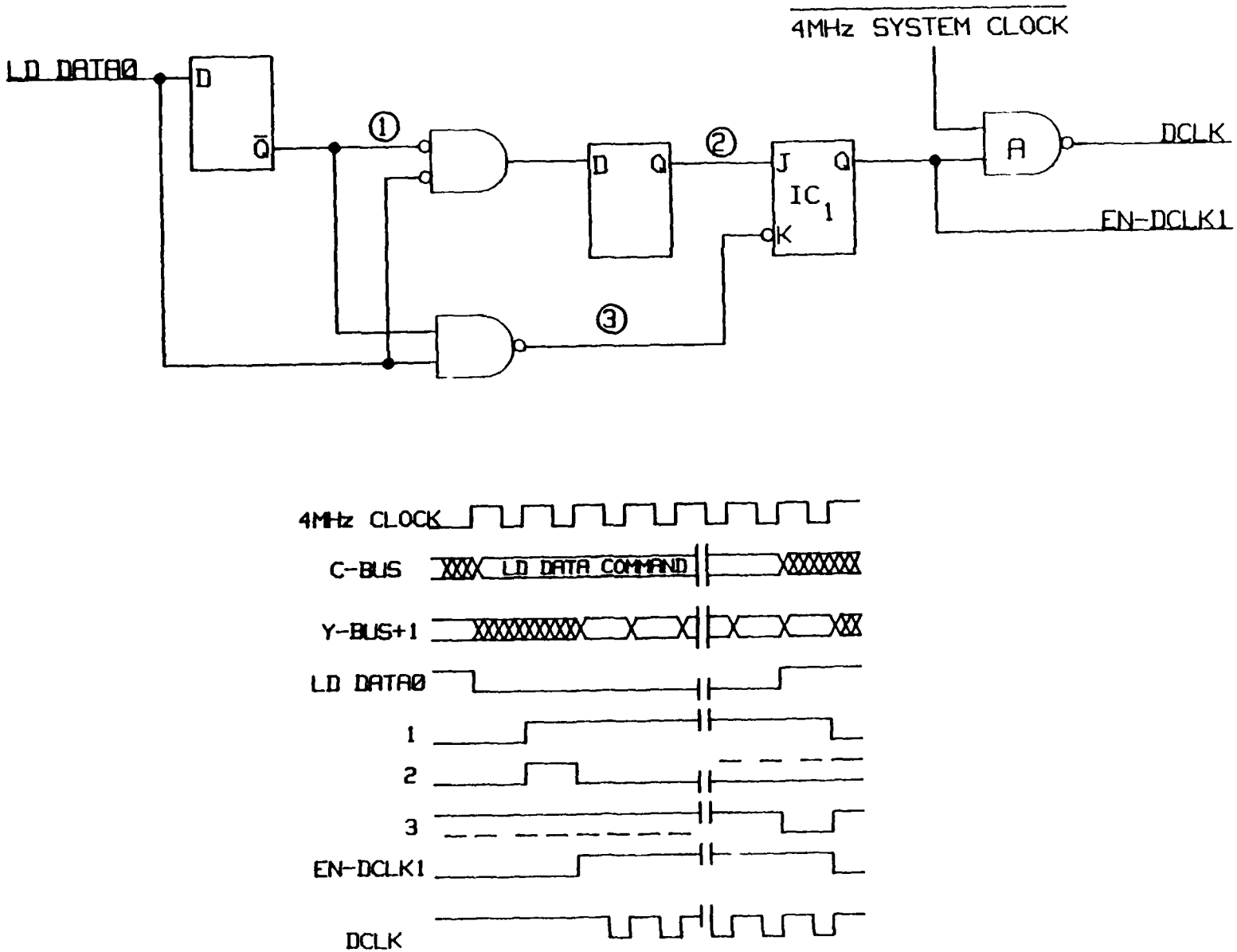


FIGURE 6.8.2.1.4-2. DATA LOAD CONTROL LOGIC WITH TIMING WAVEFORMS

6.8.2.1.5 Frame Check Generation Logic.

The generation of the frame check sequence (FCS) is performed by the 9401 Cyclic Redundancy Check Generator/Checker. The 9401 has eight selectable polynomials, one of which is a 16^{th} order polynomial shown below;

$$X^{16} + X^{12} + X^5 + 1$$

used as the divisor of the frame to produce the binary remainder which becomes the FCS. This polynomial is used by the ADCCP.

The 9401 consists of a ROM, a 16-bit register, associated control circuitry as shown in figure 6.8.2.1.5-1. The polynomial select field S_0, S_1, and S_3 is decoded by the ROM, selecting the desired polynomial by establishing a shift mode operation on the register with Exclusive OR gates at appropriate inputs. To generate the check bits, the data stream is entered via the Data Input (D) line, using the high-to-low transition of the clock on the Clock Input (CP) line. The data is gated through the most significant output (Q) of the register, and controls the Exclusive OR gates. While the data is being entered, the Check Word Enable (CWE) is held high. After the last data bit is entered, the CWE is brought low and the check bits are shifted out of the register and appended to the data bits using multiplexer M_1 as shown in figure 6.8.2.1.5-2.

6.8.2.1.6 Zero-Bit Insertion Logic.

The sequence of six consecutive 1 bits is prevented from occurring randomly within a message through the process called "Zero Bit Insertion." This process guarantees transparency by inserting a 0 bit after any five consecutive 1 bits. This is stripped by the receiver before the message is read. The logic that performs this function is shown in figure 6.8.2.1.5-2.

The 1's-bit counter is preset to a value of 10. When the data and FCS fields are being shifted in the message stream, control signal MESS/IDLE SELECT is high allowing the counter to increment each time a 1 bit is detected or set back to 10 each time a 0 bit is detected.

Hypothetically, assume the message transmission was initiated with the data pattern shown in figure 6.8.2.1.6-1. Prior to the message, the control signal MESS/IDLE SELECT is low forcing the 1's-bit counter to remain at a count of 10. When dissemination of the message begins, the control signal goes high enabling the counter to increment each time a 1 bit is seen on the serial message stream. When less than five consecutive 1's occur, the counter is set back to 10 on the following 0 bit before a ripple carry is produced. No insertion takes place. This holds true until five consecutive 1 bits occur. The counter steps from 10 to 15 at which time a ripple carry is produced. This pulse generates the active low signal SKIP0 which causes the controlled clock to skip one clock cycle essentially pausing the message stream from being shifted for that period. At the same time, the SKIP0 is used to pass a 0 bit on the transmission stream via multiplexer M_2. Also, the 1's bit counter is set back to 10.

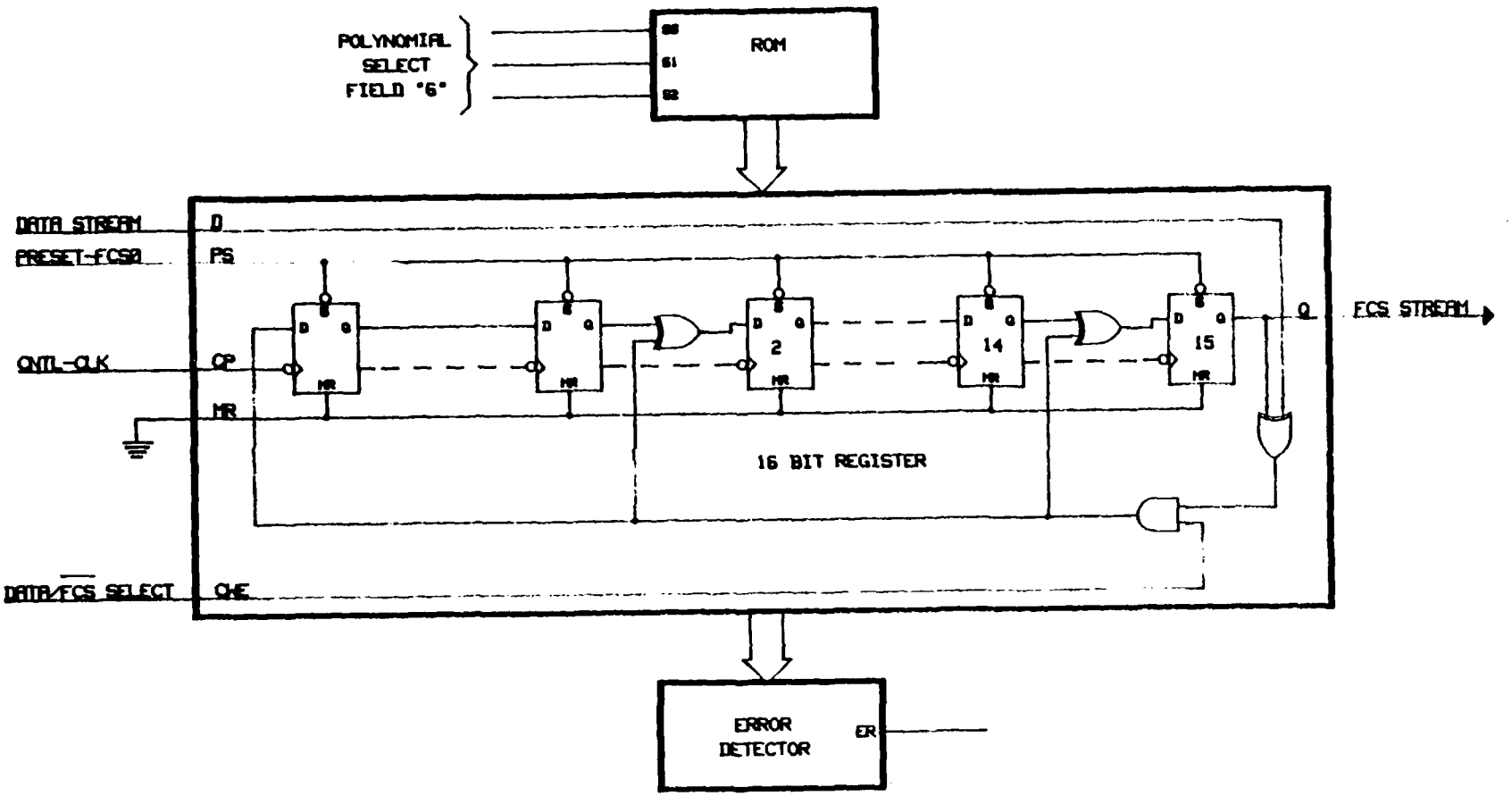


FIGURE 6.8.2.1.5-1. FCS GENERATOR/CHECKER BLOCK DIAGRAM

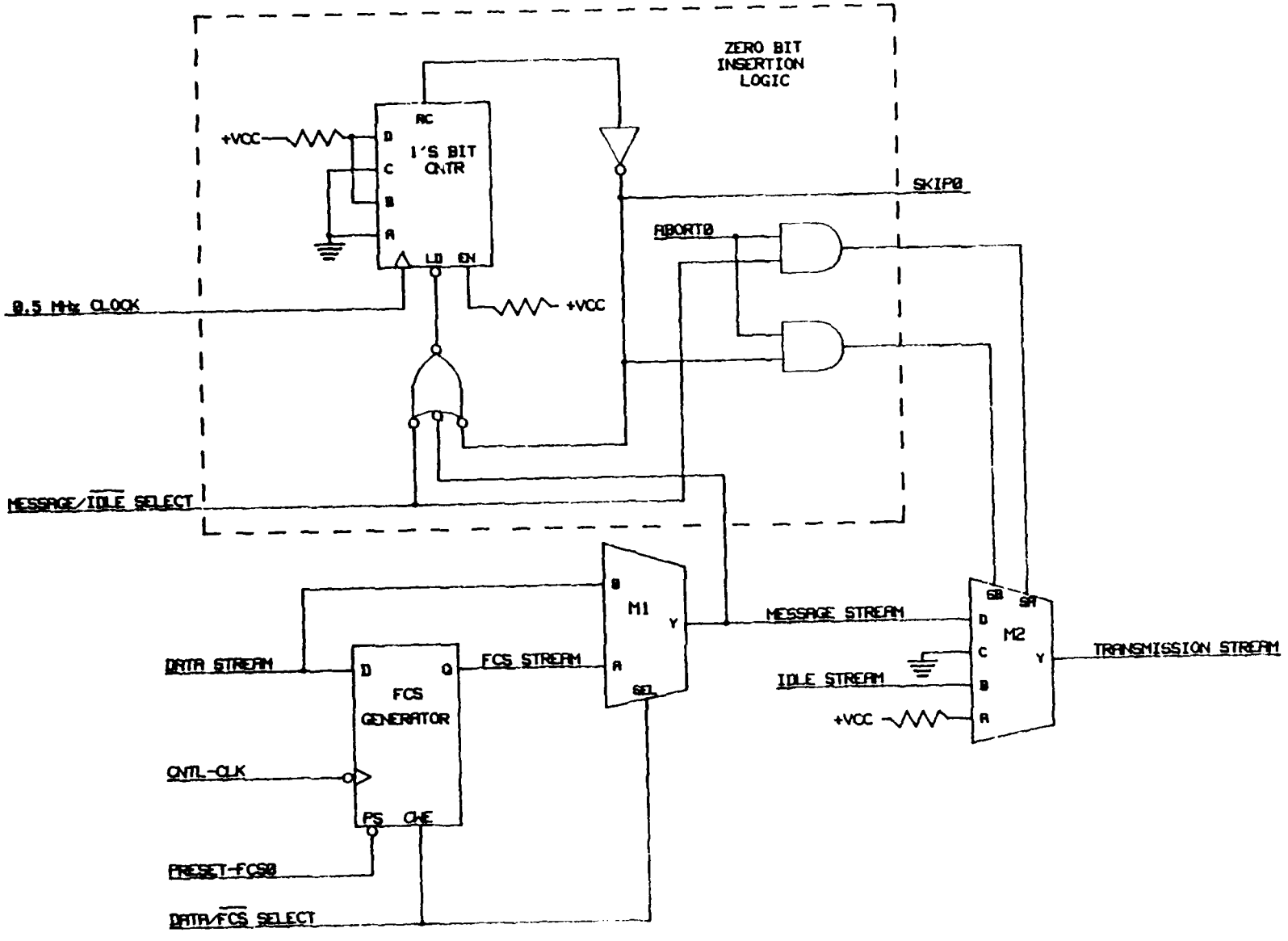


FIGURE 6.8.2.1.5-2. FCS GENERATION AND ZERO-BIT INSERTION LOGIC

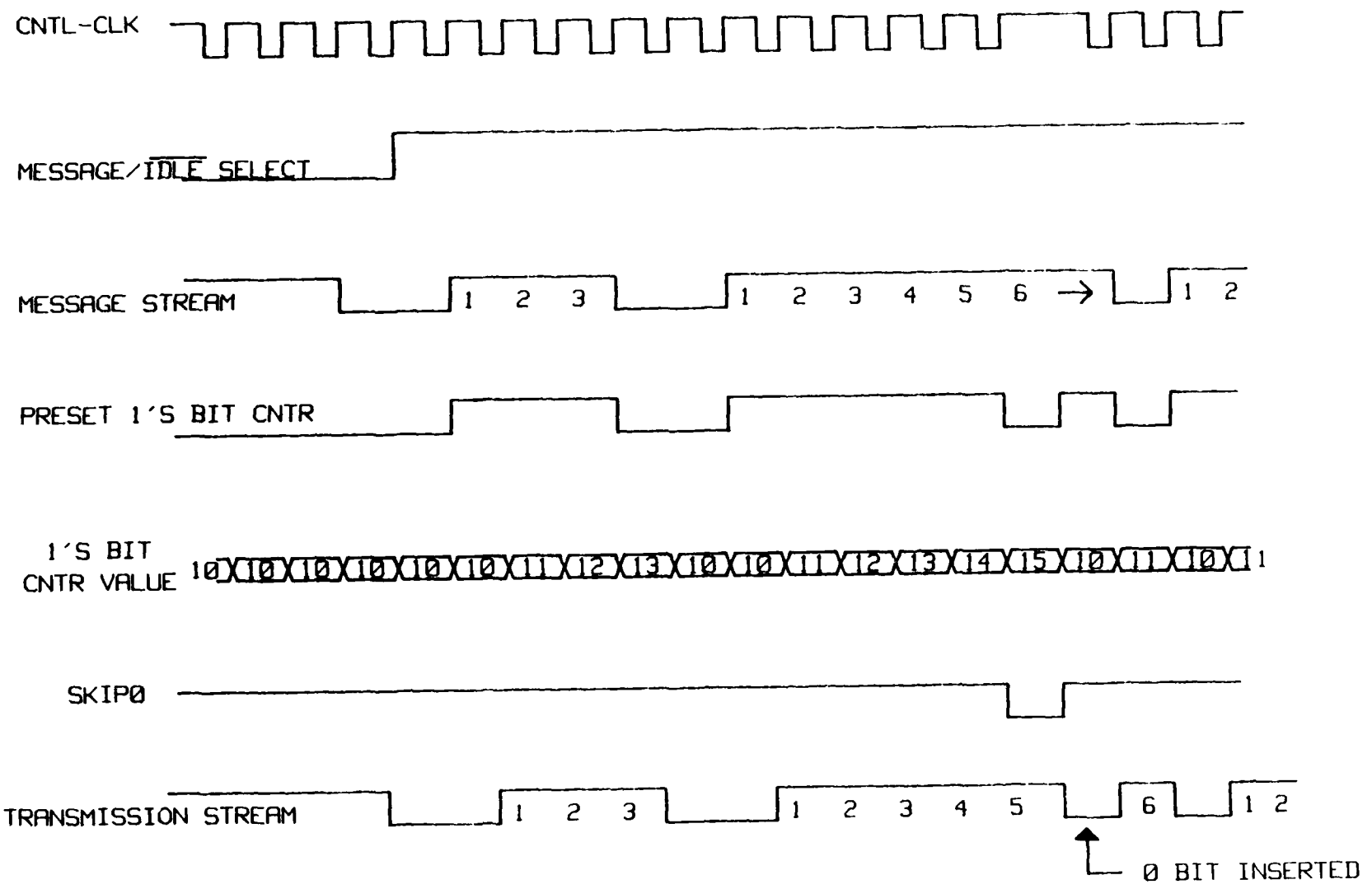


FIGURE 6.8.2.1.6-1. ZERO-BIT INSERTION TIMING WAVEFORMS

6.8.2.1.7 ADC State Controller.

Message transmission is controlled by the ADC State Controller. The controller is responsible for fetching data from the Input Buffer and inserting it between idle frames in the transmission stream. It controls the FCS generator appending the FCS field to the report and controls the 0-bit insertion circuitry. The ADC State Controller logic is shown in figure 6.8.2.1.7-1.

The controller accomplishes its tasks with the use of the condition logic shown in figure 6.8.2.1.7-2. To indicate when a report is available in the Input Buffer, the RUN flip-flop is set (or reset) each time a report is loaded into (removed from) the Input Buffer. To indicate when to insert the message in the transmission stream, the SYNC flip-flop is set (reset) when the controller is synchronized (asychronized) with the start of each idle frame in the transmission stream. To indicate when to append the FCS to the message during transmission, the comparison of the read and write address counters of the Input Buffer is used. If the counters contain the same address, the buffer is empty and the FCS is appended after the last word is shifted out of the Data Shift Register. The state diagram for the ADC State Controller is shown in figure 6.8.2.1.7-3.

To understand the operation of the ADC State Controller, consider the following action sequence; starting from synchronizing to the idle stream to the transfer of a radar report message. At this point, the following power-up condition is assumed to have just occurred:

a. The ADC is reset by the power-up state, causing the read and write address counters of the Input Buffer to be 0, the SYNC flip-flop to be in the ASYNC mode, the CYCLE flip-flop to be in the ABORT mode, and the RUN flip-flop to be in the IDLE mode. Since the SYNC flip-flop is set to the ASYNC mode, the ADC State Controller is forced into its initial state (state 0). In this state, the controller selects the 1's output through multiplexer M_2 (see figure 6.8.2.1.5-2) and attempts to synchronize to the idle frame looping through the Idle Shift Register. It performs the synchronization by holding the SYNC-WINDOW open until the IDLE STROBE sets the SYNC flip-flop to the SYNC mode. Once the controller is synchronized to the IDLE STROBE, a 1-clock wide SYNC-WINDOW pulse will be generated once every eight transmission clocks, aligned to the IDLE STROBE pulse, to reassure that synchronization with the idle stream is maintained.

b. The controller continues to transmit the 1's pattern until a minimum of eight 1's have been transmitted (state 1). This represents an abort (reset) command to receiving circuitry. The abort command is also used by the ADC diagnostic support circuitry discussed in section 6.8.2.2.

c. The controller then sets the CYCLE flip-flop to its CYCLE mode via control signal END-ABORT0 and selects the idle frames through multiplexer M_2 by removing the control signal ABORT0 (state 2). The controller will remain in the idle state (state 2) as long as no report is available for transmission.

d. To prevent the ADC Controller from transmitting before the complete report is loaded into the Input Buffer, a unique condition signal is employed to indicate when the report is ready for transmission. This is the purpose of J/K flip-flop IC_1. Flip-flop IC_1 is used to trap the GO pulse issued by the RRC after a complete report is transferred to the Input Buffer. J/K flip-flop IC_2 is used to prevent the RUN flip-flop from being set to the RUN mode before the State Controller is ready. This is necessary because the RRC can load a complete report into the Input

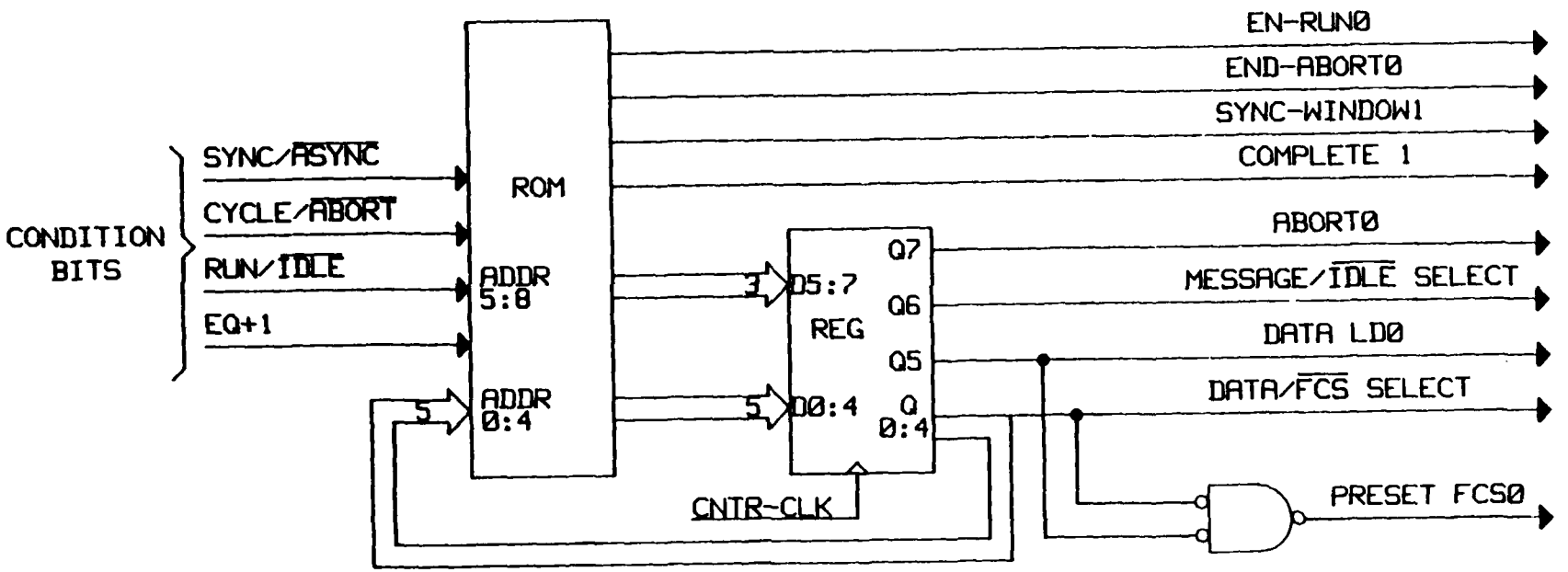


FIGURE 6.8.2.1.7-1. ADC STATE CONTROLLER BLOCK DIAGRAM

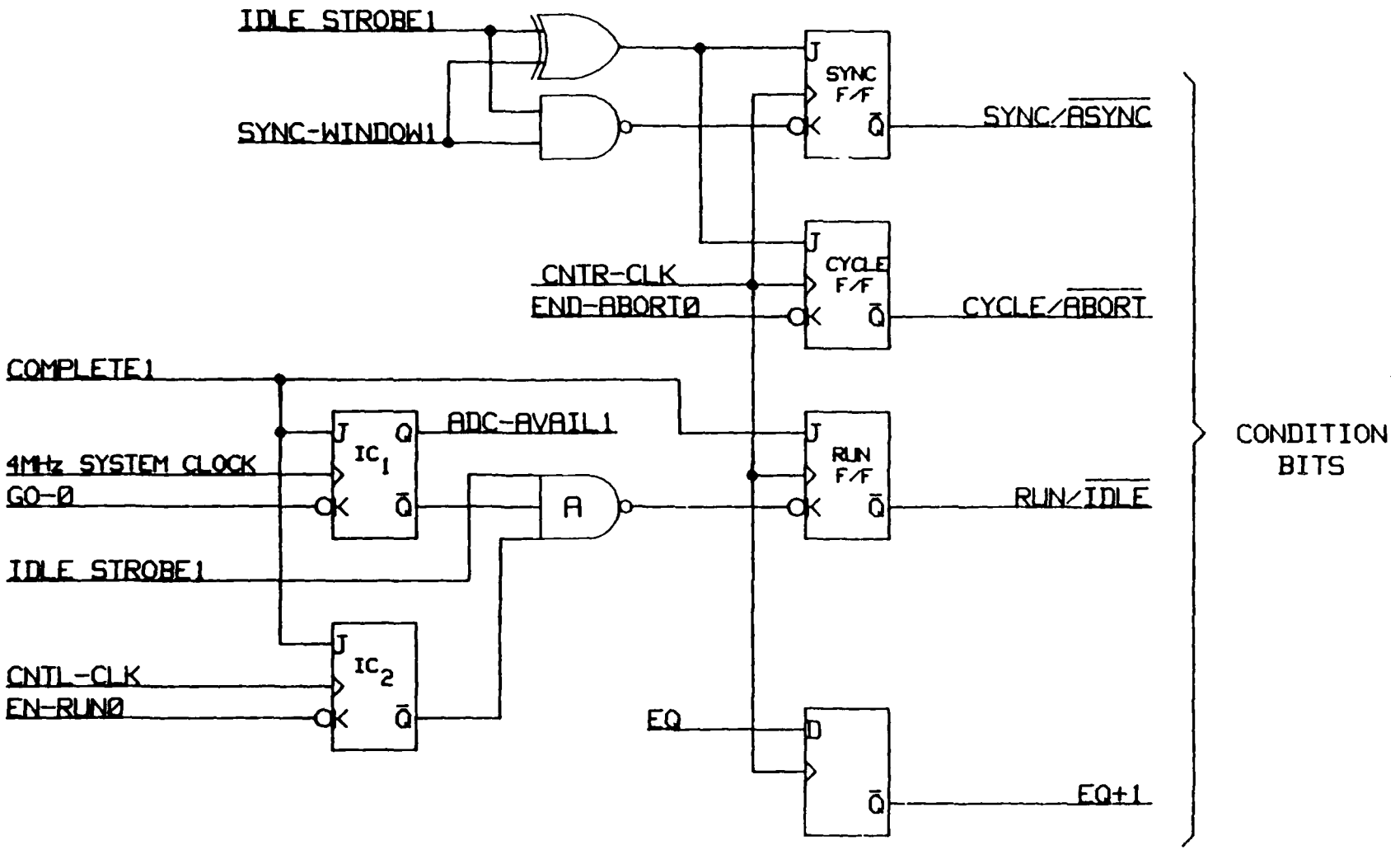


FIGURE 6.8.2.1.7-2. ADC STATE CONTROLLER CONDITION LOGIC

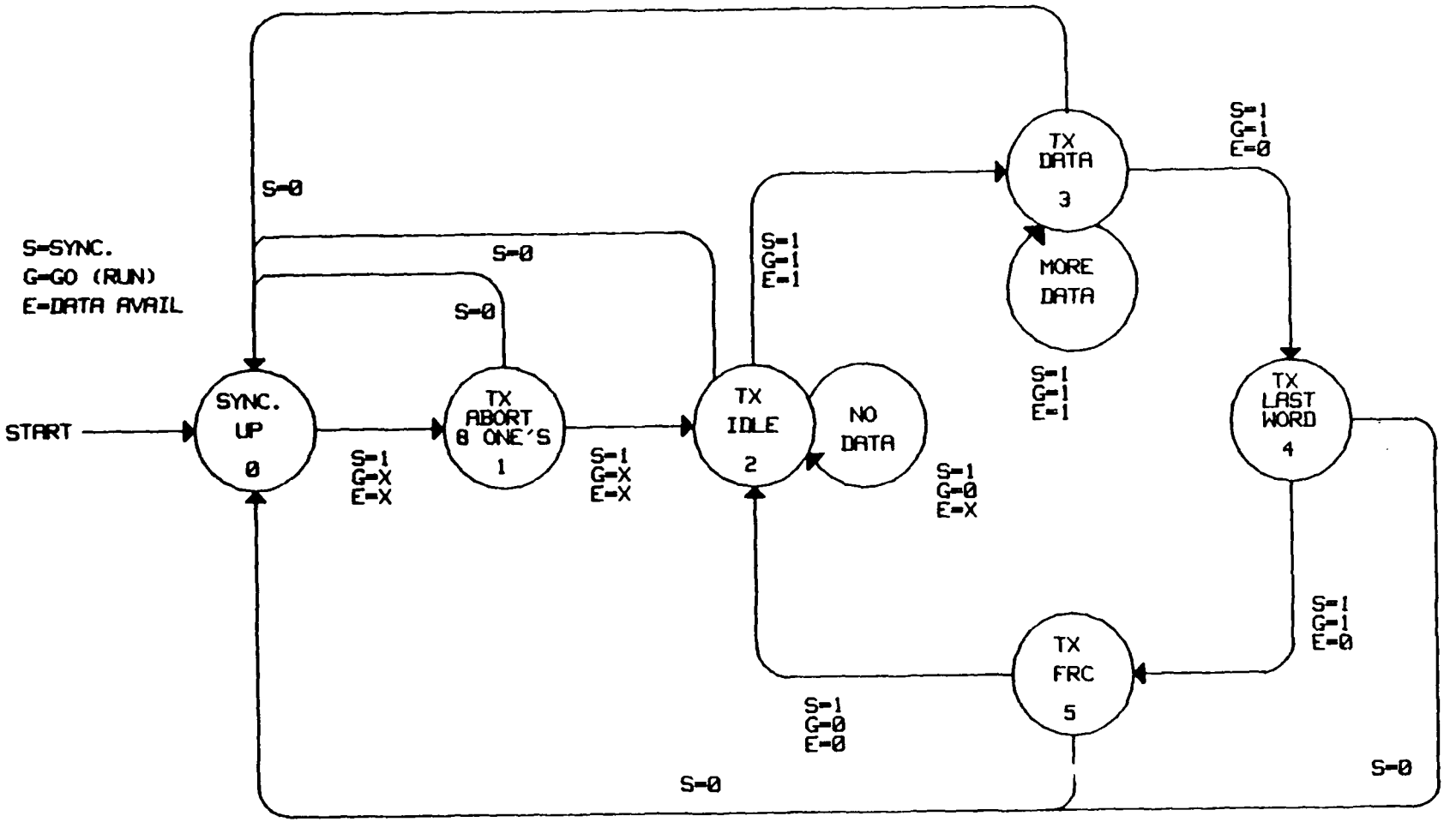


FIGURE 6.8.2.1.7-3. ADC STATE CONTROLLER STATE DIAGRAM

Buffer and issue a GO command before the State Controller has had time to link up with the idle stream. When the controller is ready it sets, flip-flop IC_2 via control signal EN-RUN0.

e. Now assume that a complete primary radar report, prefixed with the address and command fields, is transferred to the Input Buffer. When both flip-flops IC_1 and IC_2 are set, NAND gate A is enabled. When the IDLE STROBE is produced, it is passed to the K input of the RUN flip-flop setting it to the RUN mode on the following transmission clock cycle (state 3). The controller then transfers the first word from the Input Buffer into the Data Shift Register, starts the FCS generator, enables the 0-bit insertion logic, and selects the data stream through multiplexers M_1 and M_2 to the transmission stream. A timing waveform diagram is given in figure 6.8.2.1.7-4 showing the message in the transmission stream, the transmission clock and the control clock, along with several control signals produced by the ADC State Controller.

f. After 16 transmission clocks, the word transfer is completed and the read address counter is stepped to the next location. The Input Buffer is checked for additional data. The controller will remain in the data transmit state (state 3) as long as data remains in the buffer. After 16 transmission clocks, another word is loaded into the Data Shift Register. Again, the buffer is checked for additional data. This sequence is repeated until the buffer is emptied.

g. When the last word is transferred from the buffer to the Data Shift Register, the condition line EQ+1 goes high indicating that the last word of the report was just loaded into the Data Shift Register. The controller then moves to the transmit last word state (state 4). The transmit last word state prepares the controller to append the FCS field to the report. For the next 16 transmission clocks, the State Controller still holds its control lines fixed as if in the data transmit state.

h. After the last word is shifted out of the Data Shift Register, the FCS stream is selected through multiplexer M_1 to the message stream via control signal DATA/FCS SELECT going low (state 5). The State Controller remains in this state until all 16 FCS bits have been shifted on the transmission stream.

i. As soon as the FCS field has been transmitted, the State Controller selects the idle stream through multiplexer M_2 on to the transmission stream and disables the 0-bit insertion logic via control signal MESS/IDLE SELECT going low. Also the condition flip-flops IC_1, IC_2, and RUN are reset via control signal COMPLETE1 preparing the State Controller for the next message transmission cycle. The controller is now back to the transmit idle state (state 2).

j. State 2 through 5 are repeated for all following messages transmitted.

Note if synchronization with the idle stream is lost, a hardware reset is produced, clearing the Input Buffer and forcing the ADC State Controller back to its initial state (state 0). Also note that the CNTL-CLK is used to clock the ADC State Controller. This is necessary because the message stream must be paused for 1 clock whenever a 0 bit is inserted in the transmission stream.

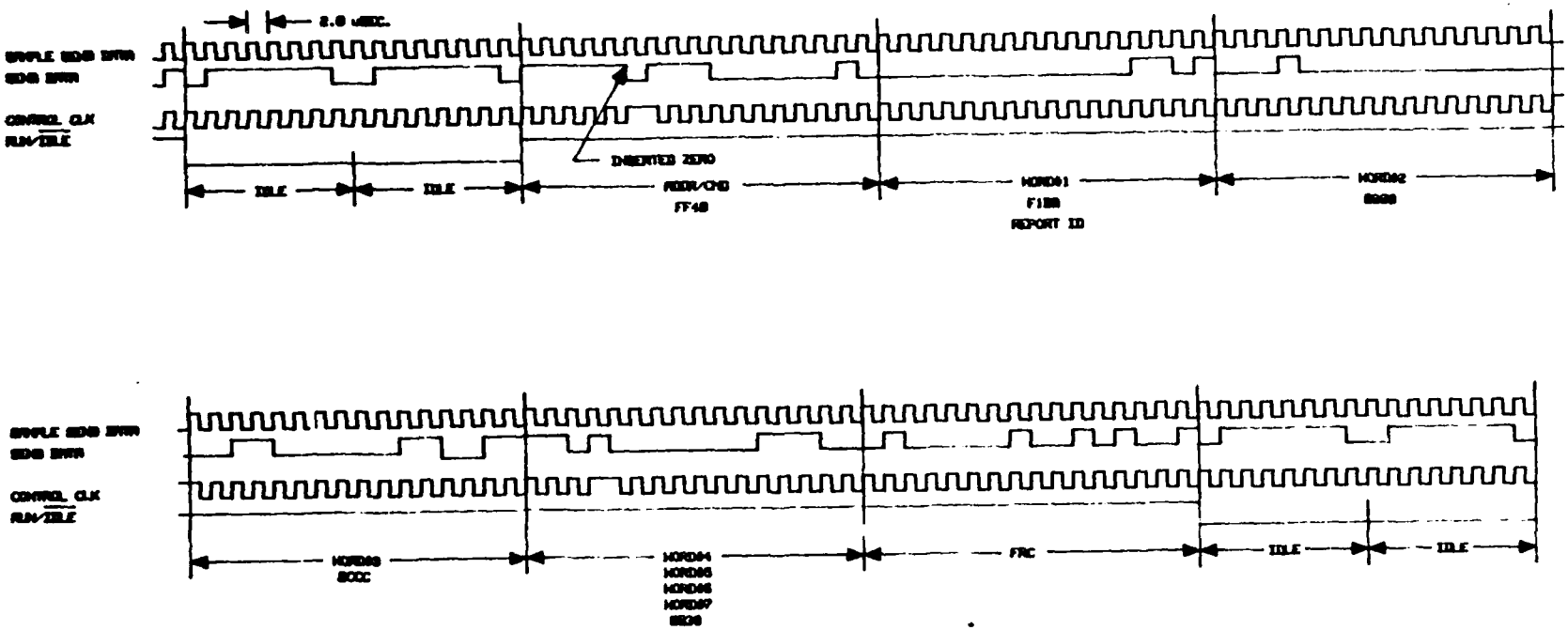


FIGURE 6.8.2.1.7-4. ADC MESSAGE TRANSMISSION TIMING WAVEFORMS

6.8.2.2 ADC Diagnostic Circuitry.

Figure 6.8.2.2-1 shows the major elements of the ADC Diagnostic Circuitry which consists of the following:

a. Receiving Logic
b. Output Buffer
c. Message Block Assembler

This diagnostic circuitry was integrated into the ADC to provide the capability of verifying the proper transmission of simulated primary radar data conforming to ADCCP. It accomplishes this by sampling the output from the ADC transmission circuitry, extracting messages and returning them to the computer via the RRC for analysis. These messages are made up in part by the original reports generated by the computer. A command and address frame are prefixed to the reports by the RRC and the FCS frame is appended to the reports by the ADC during transmission. Also, independent verification of the FCS field is performed.

6.8.2.2.1 Receiving Logic.

The receiving logic consists of two shift registers and decoding gates shown in figure 6.8.2.2.1-1. The received data is shifted into the 13-bit serial-in-parallel shift register, REG_1. This register is clocked using the received transmission clock. As the received data is shifted through the REG_1, idle and abort frames are detected via NOR gate A and NAND gate B, and NAND gate C, respectively. The received data is taped eight clocks into REG_1 and fed to a second shift register, REG_2. This is necessary to allow the control logic time to detect a report and begin extraction. Actually, the start of a report is assumed when an idle frame is not detected, therefore, requiring the first eight bits of REG_1 be checked for the existence of idle frames. REG_2 is clocked by the received clock (SKIPCLK) which is controlled to extract inserted 0 bits from the data stream. This clock is controlled by NOR gate D and NAND gate E which sample the last seven bits of REG_1 looking for a binary pattern of "0111110" when receiving a message. Note that the last "0" is not clocked into REG_2 when this pattern is detected.

6.8.2.2.2 Output Buffer.

The Output Buffer is a 16-bit x 16-word memory constructed of four 4-bit wide FIFOs. The input of the buffer is partitioned into an upper and lower half and both share a common bus with REG_2. The output of the buffer is placed on the D-Bus.

The RRC fetches data from the output buffer by placing a read data command on the C-Bus. This command is decoded by the ADC Command Decode Logic (see figure 6.8.2.1.4-1) to produce the active low read data pulse, RD DATA0. This pulse is clocked through two D flip-flops, the outputs of which are used to place the output of the buffer on the D-Bus and read out the data from the buffer, respectively.

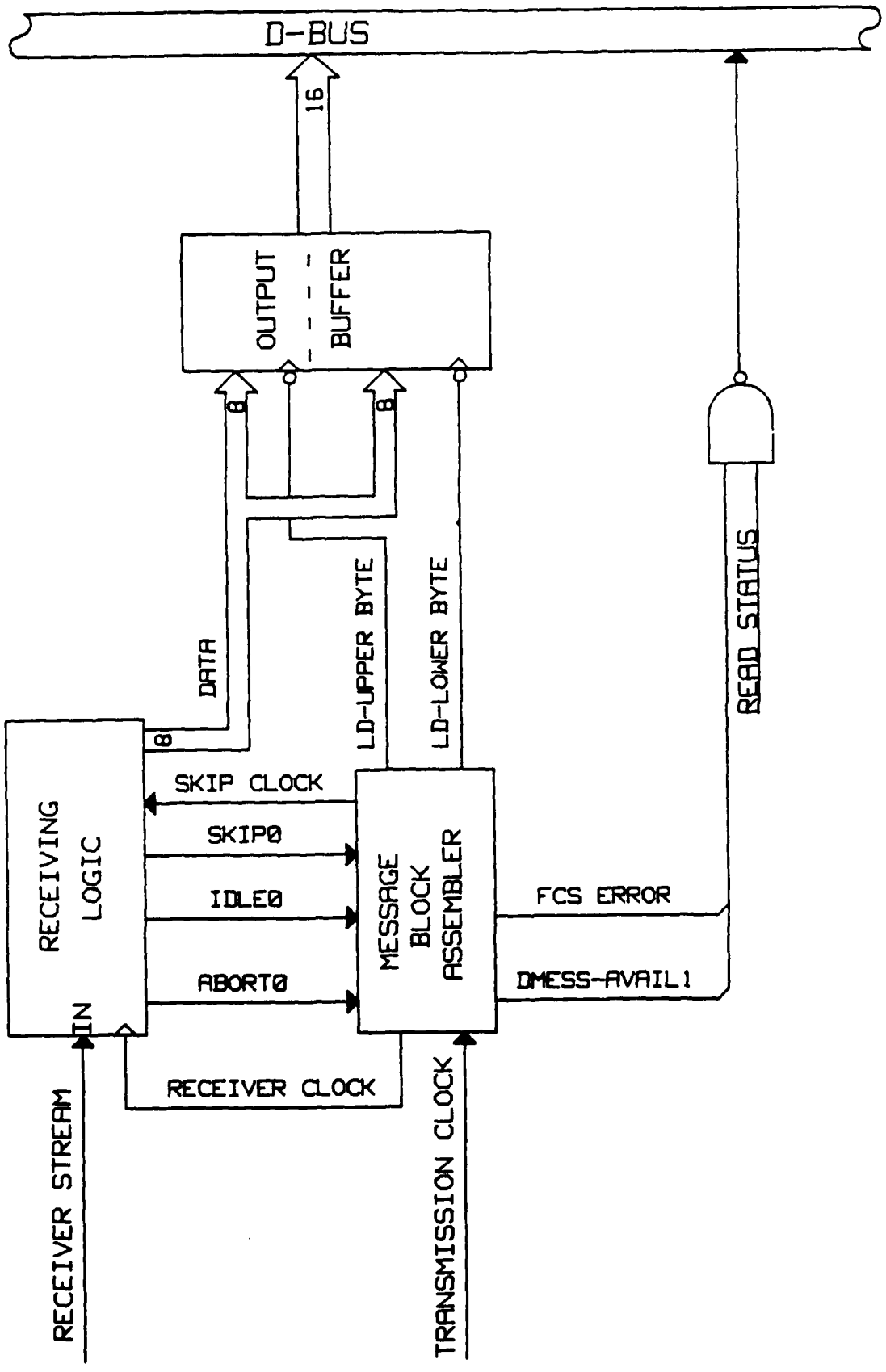


FIGURE 6.8.2.2-1. ADC DIAGNOSTIC CIRCUITRY BLOCK DIAGRAM

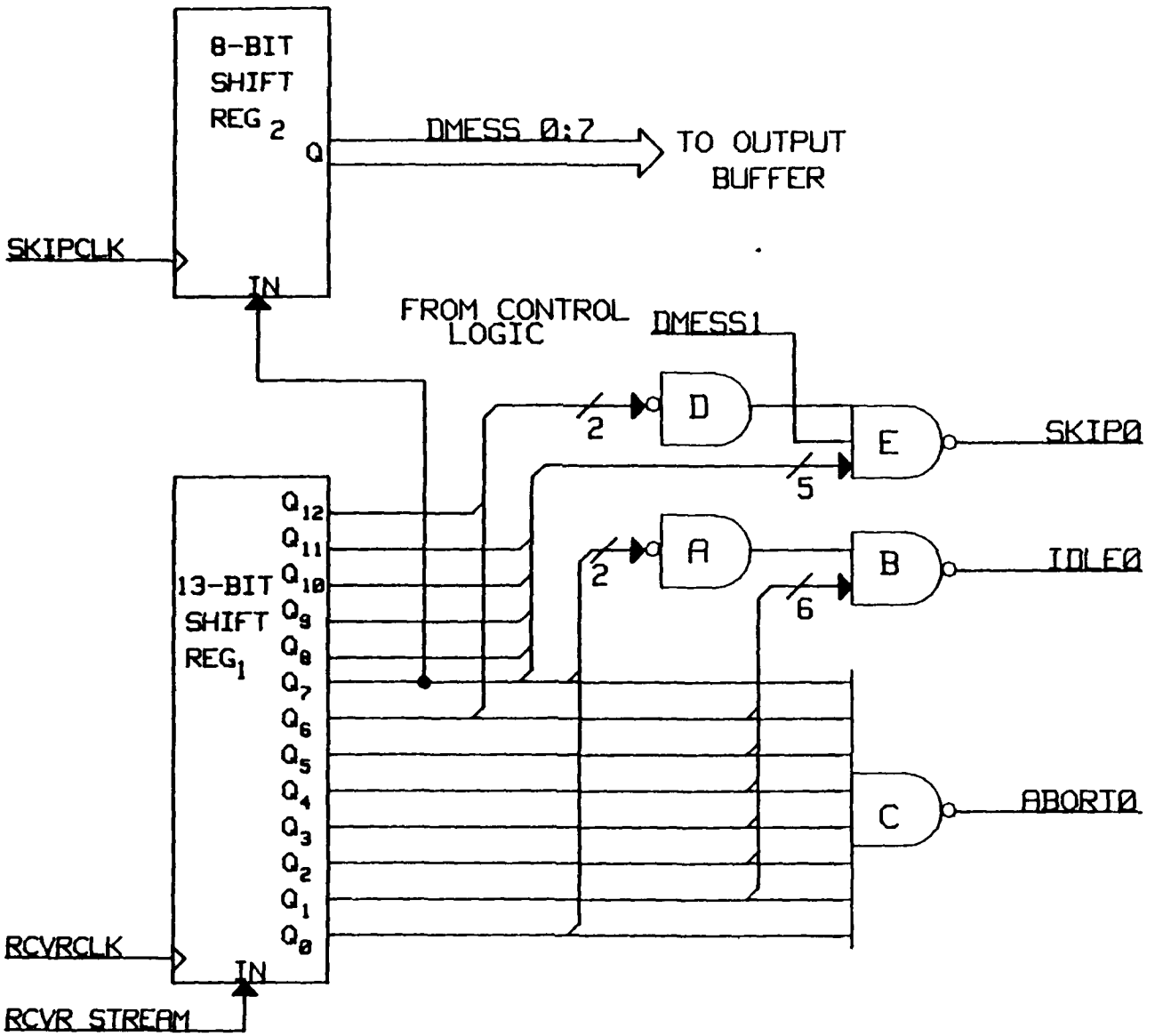


FIGURE 6.8.2.2.1-1. ADC DIAGNOSTIC CIRCUITRY RECEIVING LOGIC

6.8.2.2.3 Message Block Assembler.

The ABORT0, IDLE0, and SKIP0 pulses from the receiving logic, as well as EMPTY0 signal from the Output Buffer logic, are used by the Message Block Assembler to perform several functions. These functions are (1) to detect incoming messages, (2) remove all 0 bits inserted into the messages, and (3) reassemble the messages in the Output Buffer for transfer to the RRC.

The operations of the Message Block Assembler may be understood by considering the following (refer to figure 6.8.2.2.3-1 concerning the following discussion). Assume an abort frame followed by a series of idle frames is detected. This causes the following events to happen. After the abort pulse is produced, the Data Available flip-flop IC_1 is set to its no data state, the Message-in-Progress flip-flop IC_2 is set to its idle state, and the Idle Locked-On flip-flop IC_3 is set to its unlocked state. When the first IDLE0 pulse is produced, the Lock-On counter CNTR is preset to 9. Six clocks later, the counter reaches its maximum value producing a ripple carry pulse which is fed to D flip-flop IC_4. On the following clock cycle, the ripple carry pulse is clocked through IC_4 producing a pulse which will be called for these discussions, the Locked-On pulse. This pulse is aligned with the generation of the IDLE0 pulse to continue presetting CNTR when no idle frames are detected (when receiving a message) and loading the message into the output buffer a byte at a time. At this point, flip-flop IC_3 is set to its locked-on state. AND gate B is now conditionally enabled at the time an IDLE0 pulse is produced. As long as idle frames are detected, flip-flip IC_2 remains in the idle mode.

Now assume that a message is being received. After eight clock cycles, the first byte of the message is shifted into REG_1, and the Locked-On pulse is generated to correspond with the IDLE0 pulse. However, since data is now in REG_1, no IDLE0 pulse is generated leaving NAND gate B enabled when the Locked-On pulse is present at the input of gate B. This allows the Locked-On pulse to set IC_2 to its process-message state which conditionally enables AND gates C and D and activates toggle flip-flop IC_5. The purpose of flip-flop IC_5 is to allow only one of these gates to be enabled at any one time, starting with gate C. Gate C controls the loading of the upper half of the output buffer.

After the next eight clock cycles, the first byte shifted into REG_1 has been clocked into REG_2 and a new byte of data shifted into REG_1. Again, no IDLE0 pulse is produced and the Locked-On pulse causes the following actions. The contents of REG_2 is loaded into the upper half of the output buffer and flip-flop IC_5 is toggled to select the lower half of the buffer. After another eight clock cycles, the second byte of data is clocked into REG_2 and a third byte of data is shifted into REG_1. Again, no IDLE0 pulse is produced and the Locked-On pulse causes the following actions: (1) the contents of REG_2 is loaded into the lower half of the output buffer, and (2) flip-flop IC_5 is toggled back to the upper half of the buffer. This sequence is repeated as long as idle frames are not detected.

Now consider that the last byte of the message is in REG_1 and an idle frame will follow. The last byte of the message is clocked into REG_2 and the idle frame is shifted into REG_1 on the next eight clock cycles. The IDLE0 pulse is produced at the same time the Locked-On pulse is generated causing the actions that follow. The last byte of the message is loaded into the lower half of the output buffer, the Message-in-Process flip-flop IC_2 is set back to its idle state disabling AND gates C and D, and the Data Available flip-flop IC_1 is set notifying the RRC that diagnostic data is stored in the Output Buffer.

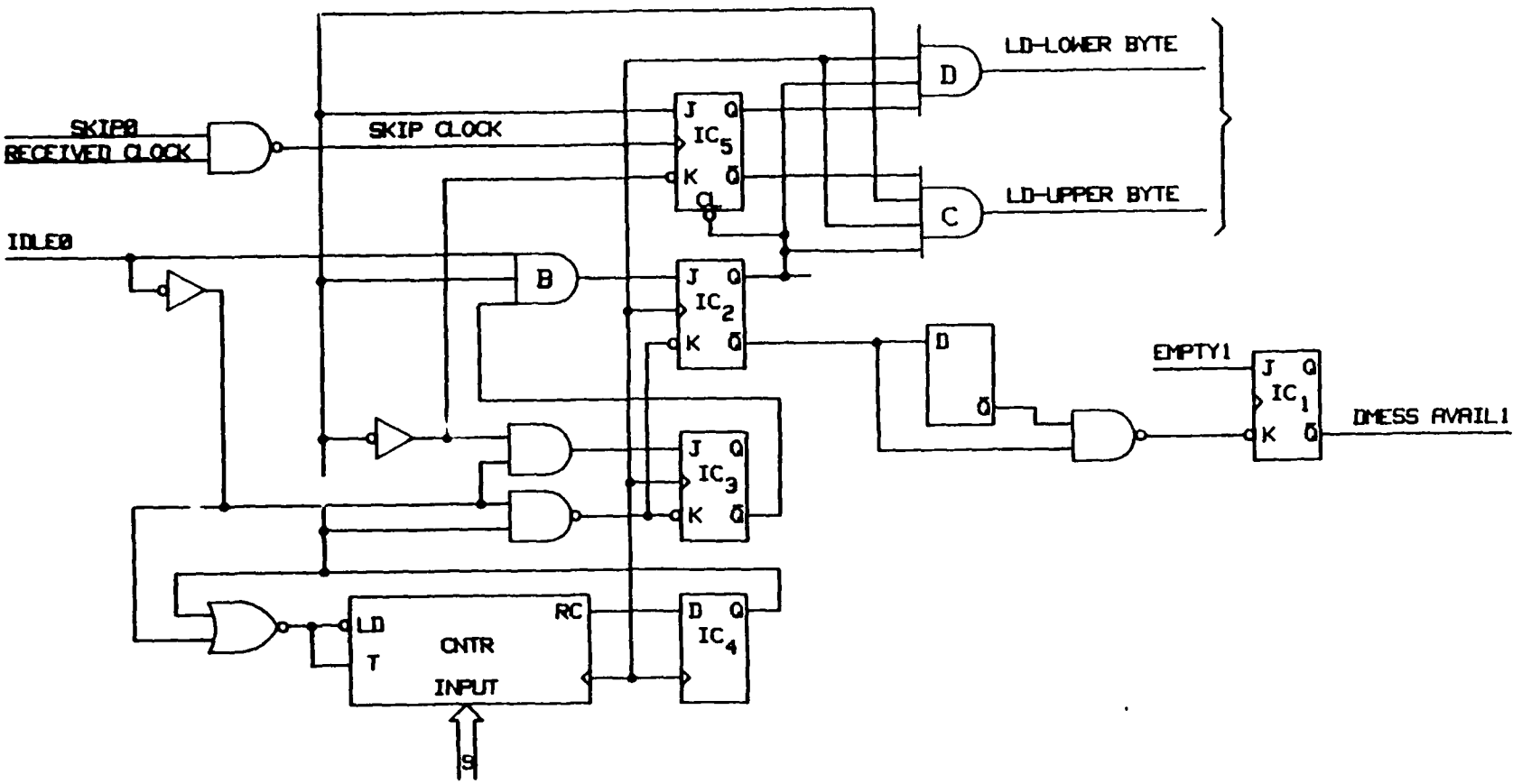


FIGURE 6.8.2.2.3-1. ADC DIAGNOSTIC CIRCUITRY MESSAGE BLOCK ASSEMBLY LOGIC

6.8.3 RRG Buffer Interface.

The RRG Buffer Interface serves as a link between the computer and the controller and provides buffer space for the computer to store radar reports as they are generated. It contains a 16-bit x 1024-word memory along with associated read/write address counters, message counter, control logic, multiplexers, and bus driver circuitry. The RRG Buffer Interface shown in figure 6.8.3-1 is almost identical to one of the buffer memory modules of the MRG Buffer Inteface, including its supporting logic.

6.8.3.1 Memory and Memory Address Logic.

The memory and memory address logic is equivalent to that of the memory and memory address logic used in the MRG Buffer Interface. Since the operations are also equivalent they will not be repeated here. (See section 6.4.1.3.1.)

6.8.3.2 Memory Write Logic.

The memory write logic shown in figure 6.8.3.2-1 is equivalent to the memory write logic used in the MRG Buffer Interface. It is a simpler version generating write control pulses for only one memory buffer. (See section 6.4.1.3.2).

6.8.3.3 Memory Read Control Logic.

The memory read control logic is equivalent to that of the memory read control logic used in the MRG Buffer Interface. Since the operations are also equivalent they will not be repeated here. (See section 6.4.1.3.3).

6.8.3.4 Message Counter and Support Logic.

The message counter is used to keep track of the number of radar reports resident in the memory buffer. During normal operations, the computer loads the message counter with the value equal to the number of radar reports loaded into the memory buffer. The controller then fetches a radar report from the memory buffer and decrements the message counter. After the report is loaded into the ADC, the controller attempts to fetch another report. When the message counter is 0, the memory buffer is empty.

Figure 6.8.3.4-1 shows the message counter and support logic consisting of:

a. An 8-bit adder
b. An 8-bit counter
c. Tristate driver logic
d. An 8-bit magnitude comparator

The 8-bit counter is an up-down type which holds the count representing the number of reports resident in memory. The output of the counter is fed to the P-inputs of the comparator. The Q-inputs of the comparator are strapped to ground representing a count of 0 (buffer empty). When the counter contains a count of 0, the output of the comparator MESS AVAIL1 is low indicating that the memory is empty. When the counter contains a number other than 0, the test signal MESS AVAIL1 is high, indicating that at least one report is resident in memory. Each time a report is read from memory, the counter is decremented by 1.

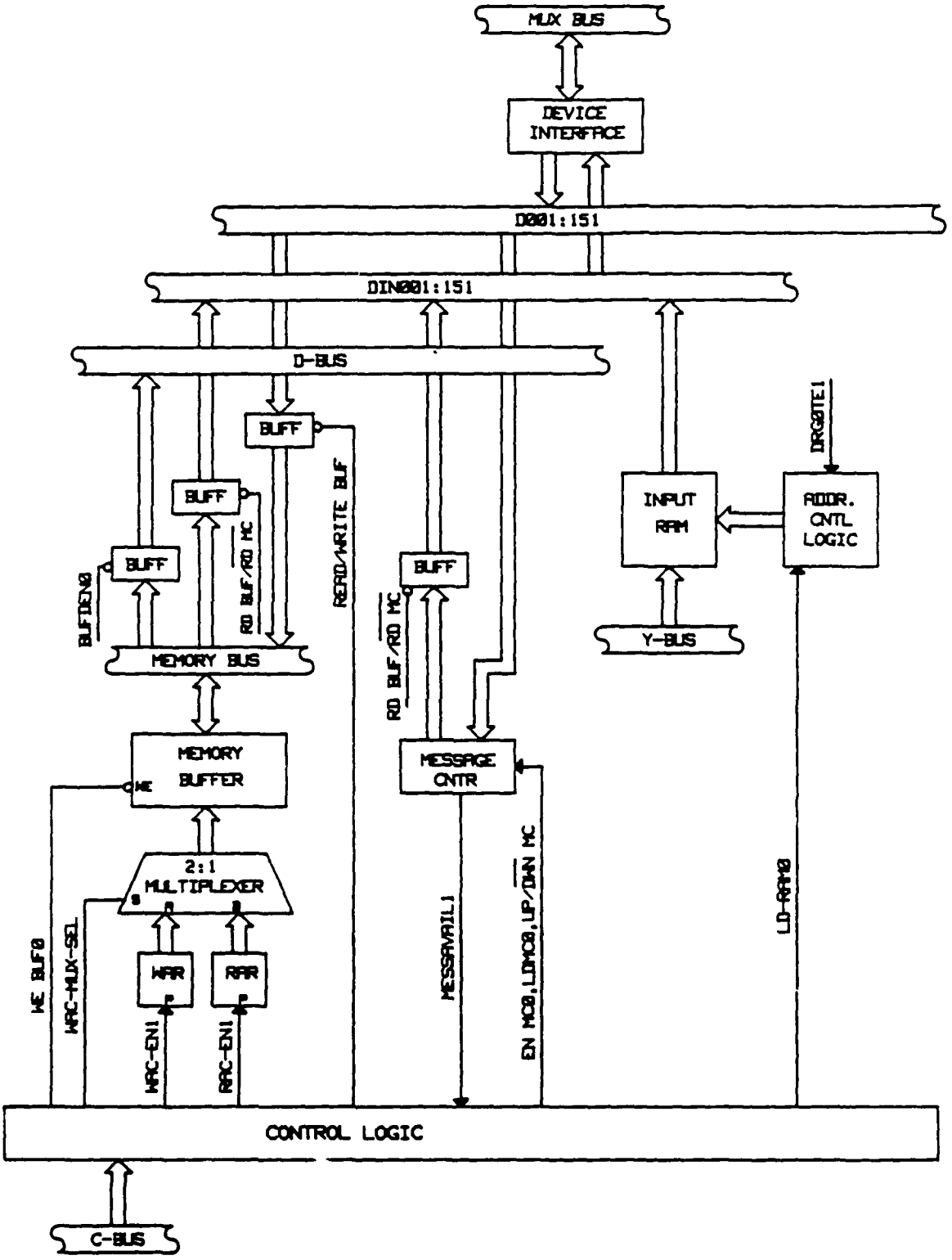


FIGURE 6.8.3-1. RRG BUFFER INTERFACE BLOCK DIAGRAM

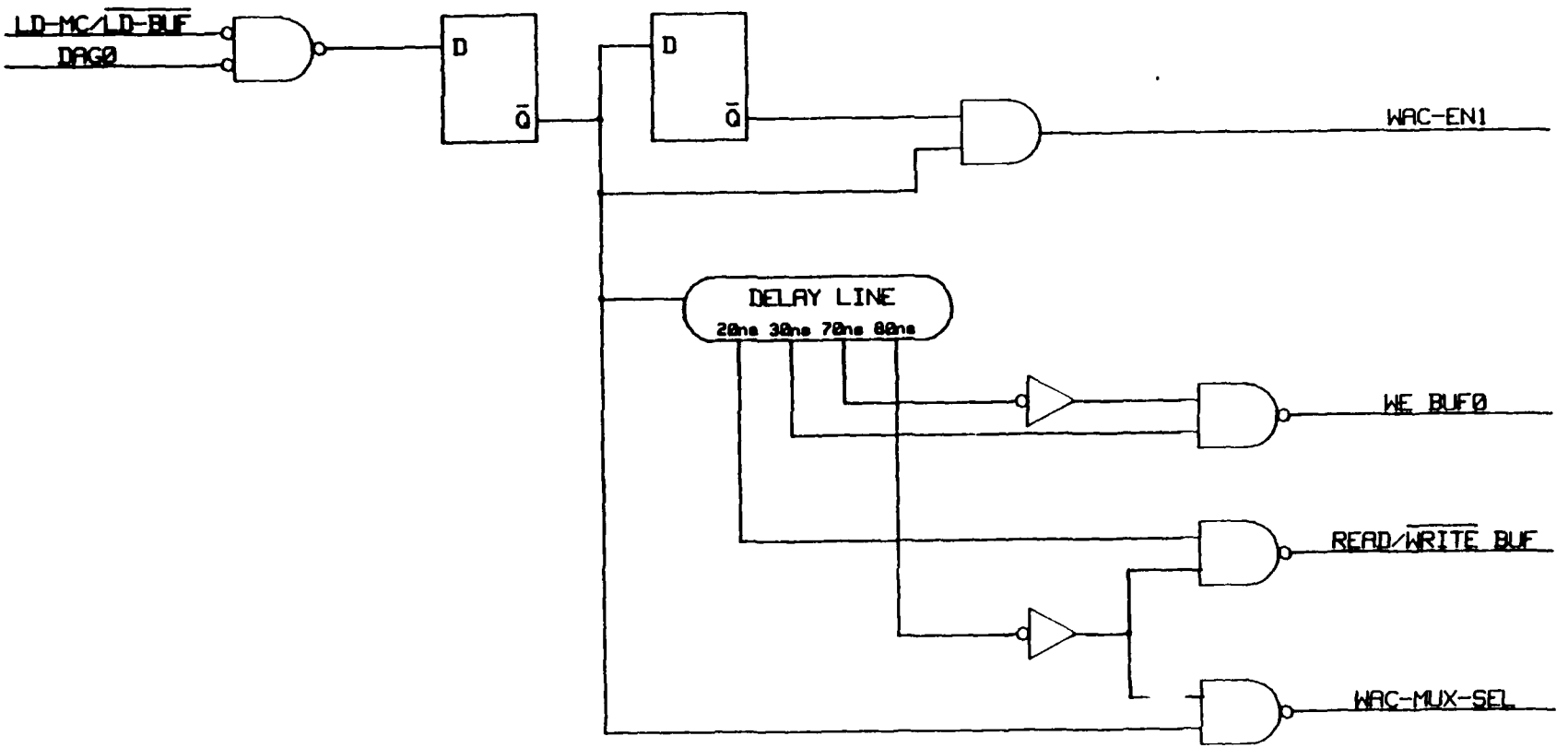


FIGURE 6.8.3.2-1. RRG BUFFER INTERFACE MEMORY WRITE LOGIC

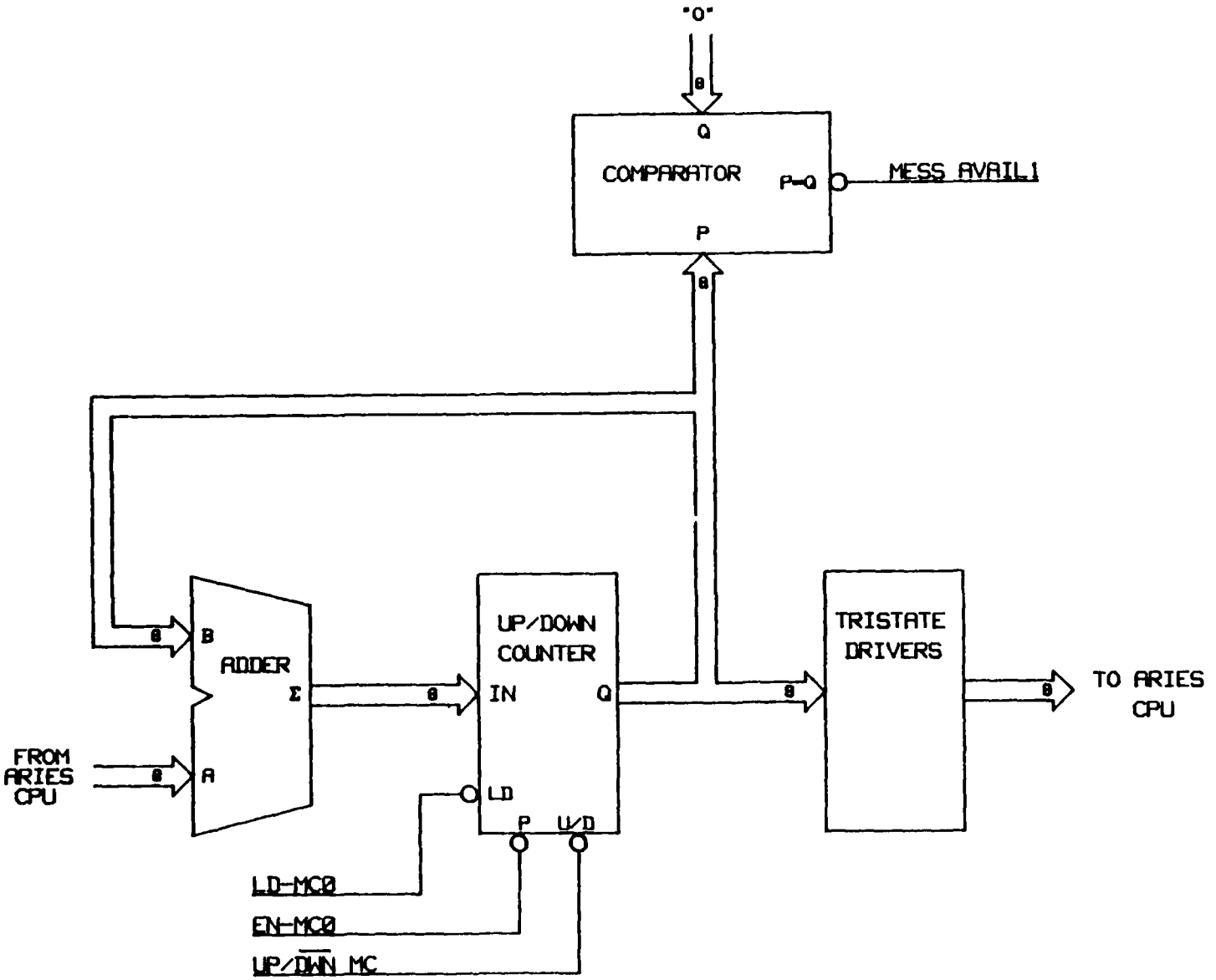


FIGURE 6.8.3.4-1. RRG BUFFER INTERFACE MESSAGE COUNTER AND SUPPORT LOGIC

To update the counter after additional reports were loaded into the memory and before all of the previous reports were completely read out, an 8-bit adder is employed. The output of the counter is fed to the B-inputs of the adder while the report update count is fed to the A-inputs of the adder. The output of the adder represents the total sum of reports currently resident in memory. This new count is then loaded into the counter.

Under diagnostic operations, the contents of the Message Counter can be placed on the D-Bus via tristate drivers. In this way, the operations of the Message Counter and supporting logic can be verified.

6.8.3.6 Operating Mode Decode Logic.

The Operating Mode Decode logic provides the means of selecting the operating mode of the RRG and controlling the various functions under each mode of operation. This logic consists of a doubled buffered register set and three single-shot pulse generators as shown in figure 6.8.3.6-1.

The doubled buffered register set is used to latch 8 bits of the asynchronous command (register A) sent by the computer, and synchronize the command bits (register B) to the 4-MHz system clock used by the RRG logic. The output of register B includes the 3-bit operation field which is fed to the 3-input decoder C that sets up the RRG Buffer Interface to the correct support configuration.

The operation field is passed to the RRC via the RRG Interface status word. The RRC uses this field to determine whether it is in the normal operating mode or in one of the diagnostic support modes as defined in table 6.8.3.6-1.

TABLE 6.8.3.6-1. RRG MODE-OF-OPERATION CONTROL FIELD

OPERATION FIELD BITS			MODE OF OPERATION
13	14	15	
0	0	0	Normal
0	1	0	ADC Loopback Test
0	1	1	Microprocessor Test
1	1	1	Buffer Interface Test

The three single-shot pulse generators produce the control pulses; RESET, DECREMENT-MESSAGE-COUNTER, and INCREMENT-READ-ADDRESS-COUNTER. When the appropriate command bit is set each time the command is issued, a single pulse is generated.

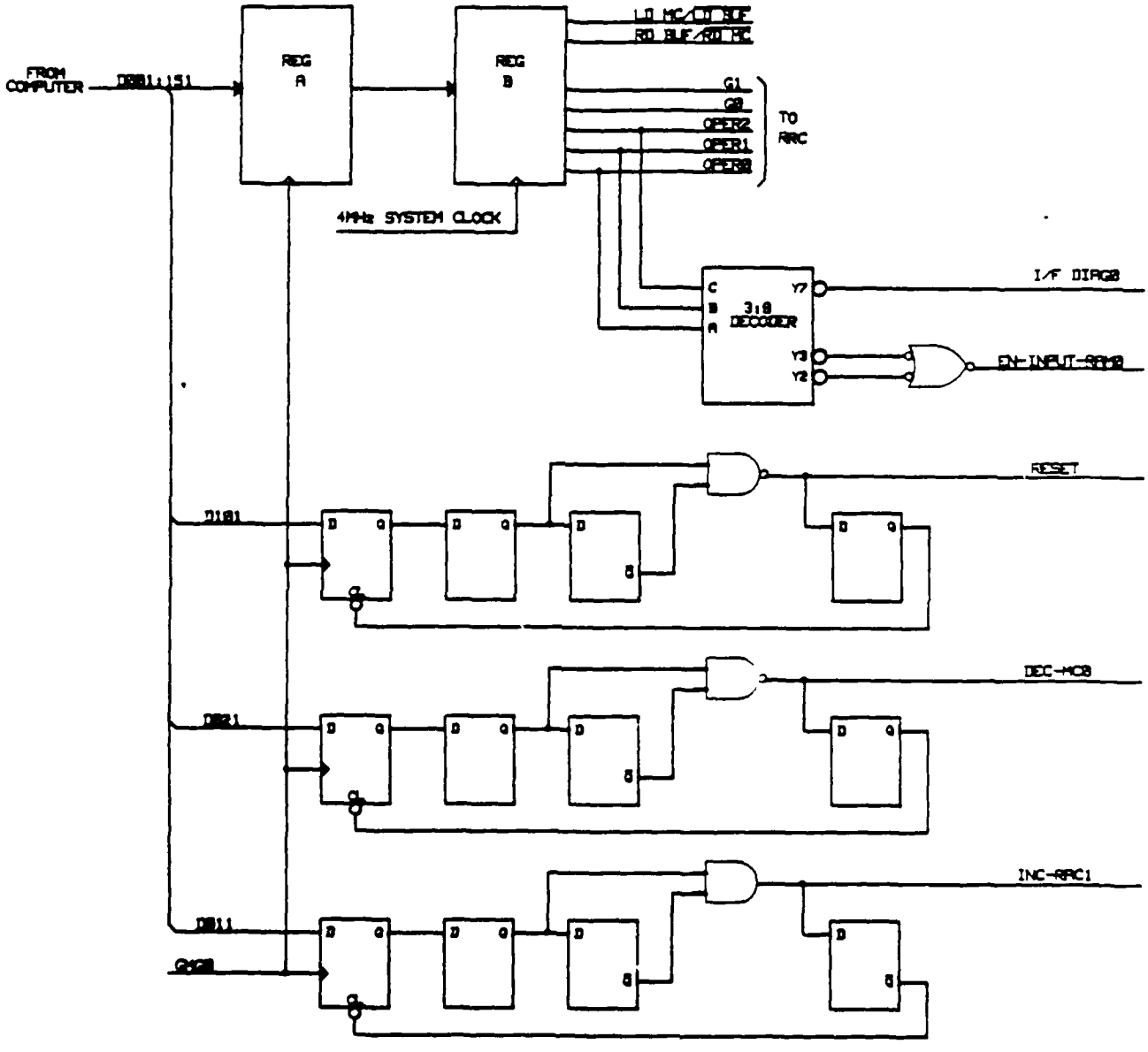


FIGURE 6.8.3.6-1. RRG BUFFER INTERFACE OPERATING MODE DECODE LOGIC

6.9 TIME-OF-YEAR RECEIVER.

The TOY Receiver performs two functions for the ARIES. First, the TOY makes available standard time data to be recorded by the ARIES data recording function. This time information is used to time correlate data recorded by the ARIES with data recorded by the Mode S sensor. Second, the TOY serves as a timer for the real-time operating software for task scheduling. Under normal operation, the TOY generates one interrupt request every 1.024 milliseconds, (ms).

The TOY Receiver is mounted on one digital board located in slot No. 3 of the ARIES Digital Chassis. The functional layout of the TOY Receiver is shown in figure 6.9-1. The TOY Receiver consists of input logic which accepts parallel transfer of TOY information and reassembles it for recording diagnostic support logic which feeds test data from the computer through the RS-422 receiver for analysis, precision timing logic, and associated control logic.

6.9.1 Input Logic.

In the normal mode of operation, the TOY Receiver accepts TOY information from the Mode S sensor. This information can be received over one of two input channels at any one time. The two TOY channels are necessary to receive information from either channel of the Mode S sensor. The channel selected is determined by the channel status signals received from each Mode S channel. The logic that determines the channel selection was covered in section 6.2.2.1.

Each TOY channel contains 12 quad RS-422 line receivers to construct a 48-bit parallel receiver interface. The output lines of both receivers are tied together forming a tristate bus 48-bits wide. The tristate bus is fed to a 48-bit input register made up of six octal D-type registers. Only one receiver is active on the bus at any one time.

The TOY information is updated in the input register once every $1/128^{th}$ of a second when the presence of new data is indicated by the active high strobe VALID. The VALID strobe is leading edge detected to produce the pulse STR-TOY-DATA0, which in turn, latches the data into the input register.

To prevent the information from being updated while being read by the computer, a second level buffer is employed. This buffer is made up of six octal D-type registers organized as a set of three registers to assemble the information into three 16-bit halfwords. The second level buffer is updated each time a request for interrupt is generated. The request for interrupt produces the LD-TOY clock which loads the contents of the 48-bit input register into the second level buffer. The TOY interrupt queues the computer to fetch the information stored in the second level buffer, allowing 1.024 ms before the next update.

6.9.2 Read Ring Counter.

Time is read from the TOY via the data request control signal DRG0. The particular word read is determined by the Read Ring Counter shown in figure 6.9.2-1. The Read Ring Counter is made up of three D-type flip-flops. The Q-output of each flip-flop is connected to the D-input of the following flip-flop to form a loop or ring.

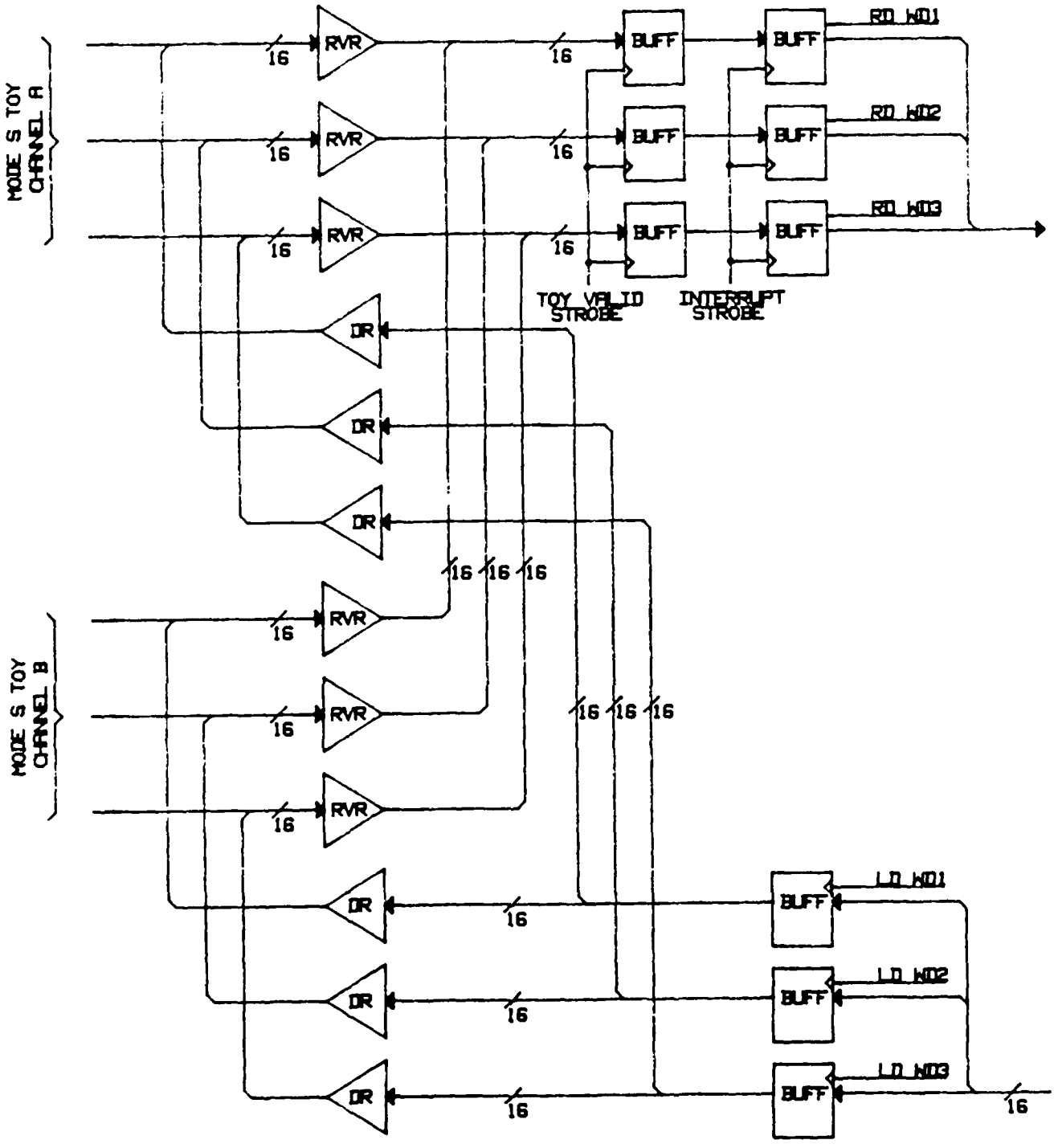


FIGURE 6.9-1. TOY RECEIVER FUNCTION BLOCK DIAGRAM

When the Read Ring Counter is reset, the binary value of 011 is loaded, corresponding to words 1, 2, and 3, respectively. The word read is specified by the active low signal. In this case, word 1 would be the first word read. The Read Ring Counter is reset each time a request for interrupt is generated or under software control via a command reset or during initial power startup. After completion of each data request, the counter is incremented producing the following sequence: 011, 101, 110, 011, etc. In this way, all three halfwords may be read starting with word 1.

6.9.3 Diagnostic Logic.

Diagnostic support logic is integrated with the TOY Receiver to provide the means of verifying the input receiver logic. This is accomplished by placing the TOY Receiver in the diagnostic mode and loading known data into the 48-bit wide output buffer. The output buffer is organized as a set of three registers to assemble three 16-bit halfwords into one 48-bit word. This word is fed to two sets of RS-422 driver channels, each consisting of 48 differential drivers. Each driver channel supplies data to a corresponding receiver channel to allow test data to be looped back to the computer through either receiver circuitry.

6.9.4 Load Ring Counter.

The Load Ring Counter is employed to control the data load sequence for the output buffer. As shown in figure 6.9.4-1, the Load Ring Counter is made up of three D-type flip-flops. The Q-output of each flip-flop is connected to the D-input of the following flip-flop to form a loop or ring.

The Load Ring Counter is initialized either by a programmed reset or during initial power startup. When initialized, the counter will contain the binary value of 011 which corresponds to the three registers making up the output buffer. The register loaded is specified by the bit that is active low. In this case, register 1 would be the first loaded. The first, second, and third halfwords will be loaded into registers 1, 2, and 3, respectively. After completion of each data load, the counter is incremented producing the following sequence: 011, 101, 110, 011, etc. In this way, three halfwords may be loaded into the output buffer, starting with register 1.

6.9.5 Digital Millisecond Timer.

Figure 6.9.5-1 shows the Digital Millisecond Timer logic. The digital ms timer is a 12-bit counter made up of three 4-bit binary counters. The timer is clocked at the rate of 4 MHz to generate a ripple carry pulse once every 4096 clock cycles. This corresponds to the generation of a pulse every 1.024 ms, hence the term digital ms timer.

The digital ms timer is held at 0 if the TOY's GPI is in the disarm interrupt state preventing the generation of ripple carry pulses. When the timer is enabled, the ripple carry pulses are generated and used to request interrupts. If the GPI is in the disable interrupt state, interrupts are pending. The ripple carry pulses are also used to reset the Read Ring Counter, and update the TOY data in the second level input buffer.

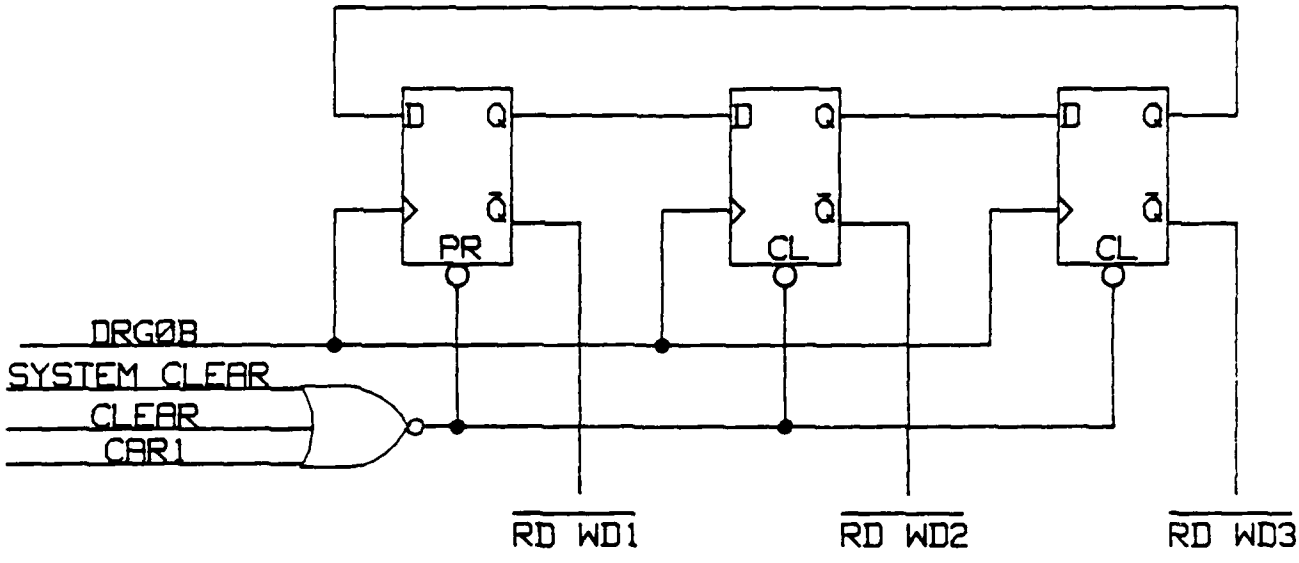


FIGURE 6.9.2-1. READ RING COUNTER

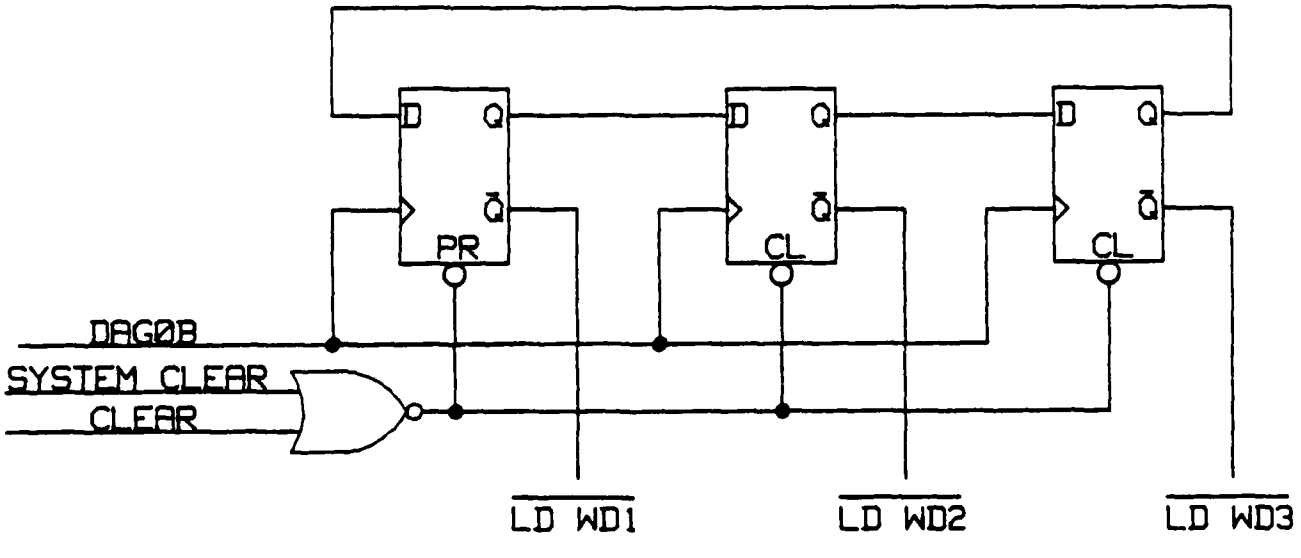


FIGURE 6.9.4-1. LOAD RING COUNTER

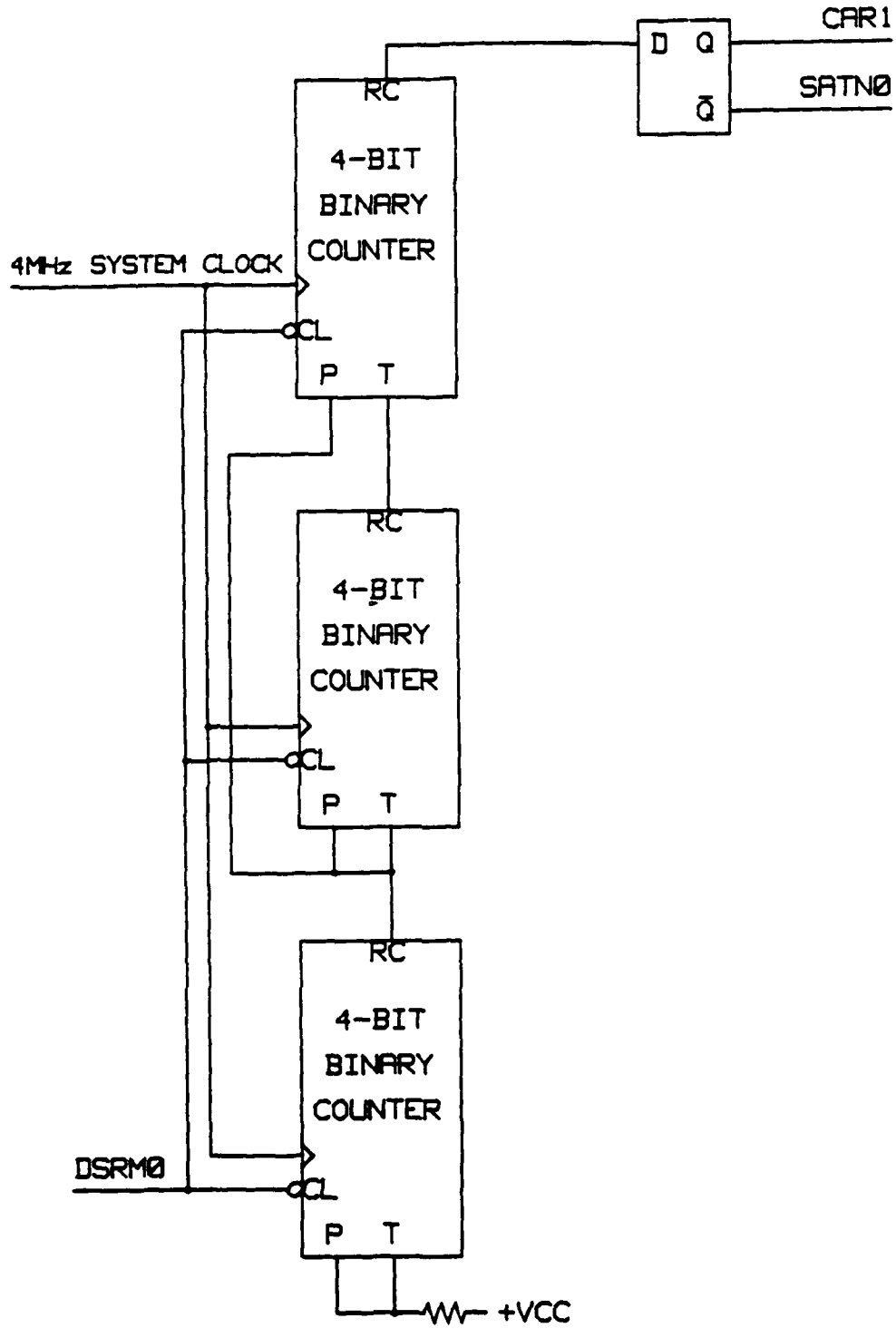


FIGURE 6.9.5-1. DIGITAL MILLISECOND TIMER LOGIC

6.9.6 TOY Control Logic.

The TOY control logic primarily controls the diagnostic operations of the TOY. It consists of two signal-to-pulse generators used to produce the CLEAR (Programmed Reset) and LOAD pulses, and a command register used to latch two diagnostic control bits. Control bit 15 places the TOY in the normal mode of operation (Bit 15 = 0) or in the diagnostic mode of operation (Bit 15 = 1). In the diagnostic mode of operation, the normal channel selection is overrode and control bit 14 is used to select the channel to be tested. When control bit 14 is low, channel A is selected and when control bit 14 is high, channel B is selected. The control logic is shown in figure 6.9.6-1.

6.10 SELF TEST UNIT.

Due to the importance of ARIES to Mode S sensor testing, and the complexity of the ARIES equipment, extensive self testing capabilities where incorporated in all special purpose digital devices designed for the ARIES. These capabilities were designed to support initial equipment checkout and subsequent maintenance diagnostic assistance. These testing modes primarily excercise all of the data paths between the computer and the special purpose devices by writing data into buffers, registers, etc., and then reading the data back. Note, many of these capabilities were discussed in the previous sections. However, none of the analog circuitry can be tested in this manner. This is the primary purpose of the Self Test Unit (STU), to perform loop testing of major ARIES subsystems; Uplink Receiver, Modeled Reply Generator, and Fruit Reply Generator. The STU with its own supporting diagnostic software provides confirmation of correct system operation prior to and following simulation runs.

Figure 6.10-1 depicts the STU in relation to the ARIES. The STU is a self-contained unit using an independent processor (Concurrent Computer Corportaion 3205 32-bit series computer) with supporting hardware to test the ARIES input and output ports at the RF level. The STU communicates with the ARIES computer via a single RS-232 link used to provide initial loading and data transfers. The STU can generate at 1030 MHz any type of ATCRBS or Mode S interrogation with any data pattern. The STU can also receive at 1090 MHz, ATCRBS, and Mode S replies and gather other pertinent information to characterize the operational readiness of the reply generation circuitry.

As shown in figure 6.10-1, the STU is composed of two independent modules; the Uplink Transmitter and the Downlink Receiver. The Uplink Transmitter is discussed in section 6.10.1 and the Downlink Receiver is discussed in section 6.10.2.

6.10.1 Uplink Transmitter.

The Uplink Transmitter consists of both digital and analog circuitry. The digital transmitter logic is self-contained on one board mounted in slot No. 5 of the STU processor chassis. The analog circuitry is mounted on the STU RF Transmitter/Uplink Receiver panel located in slot No. 9 of the ARIES analog chassis. The operations of the digital circuitry is discussed in section 6.10.1.1, and the operations of the analog circuitry is discussed in section 6.10.1.2.

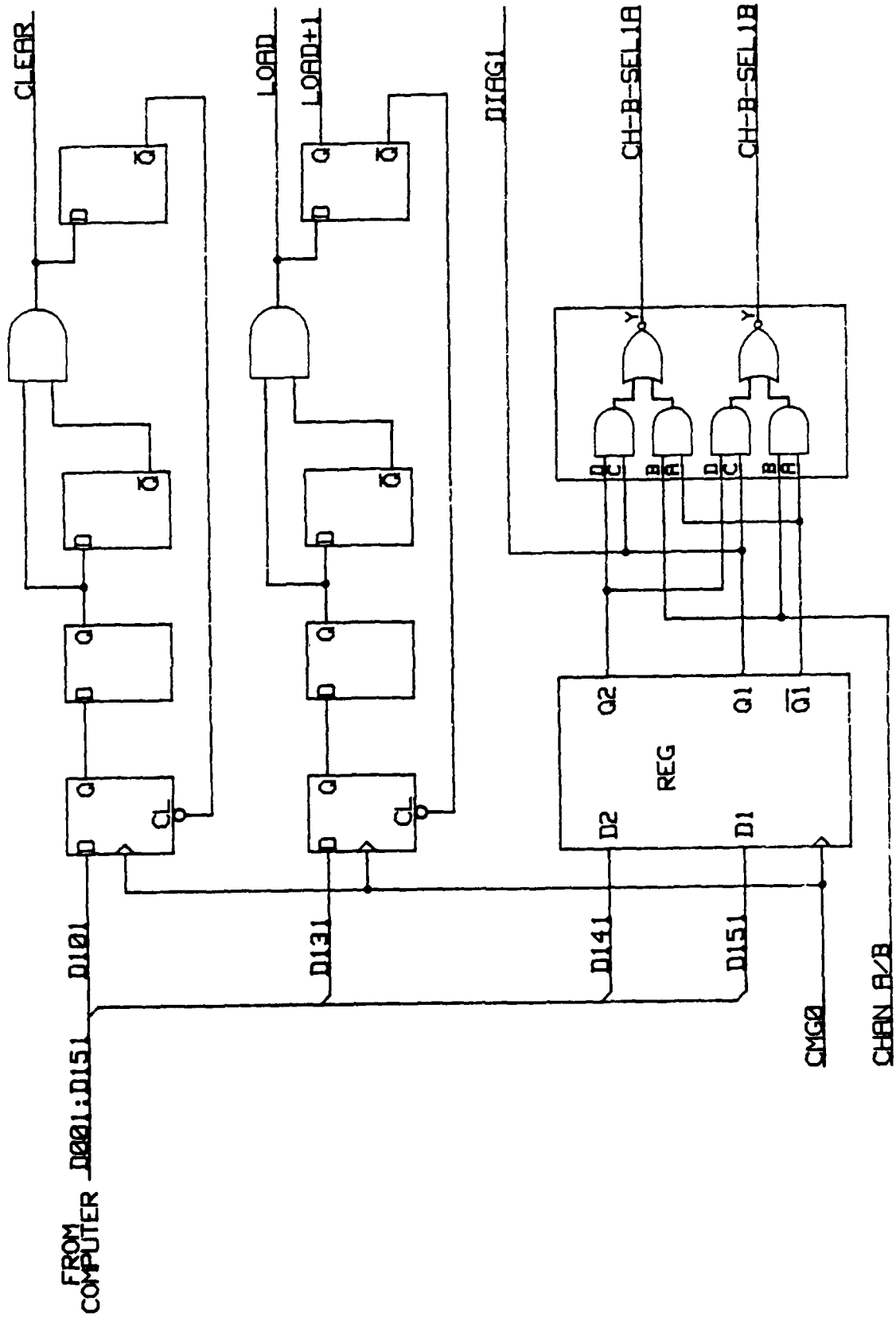


FIGURE 6.9.6-1. TOY CONTROL LOGIC

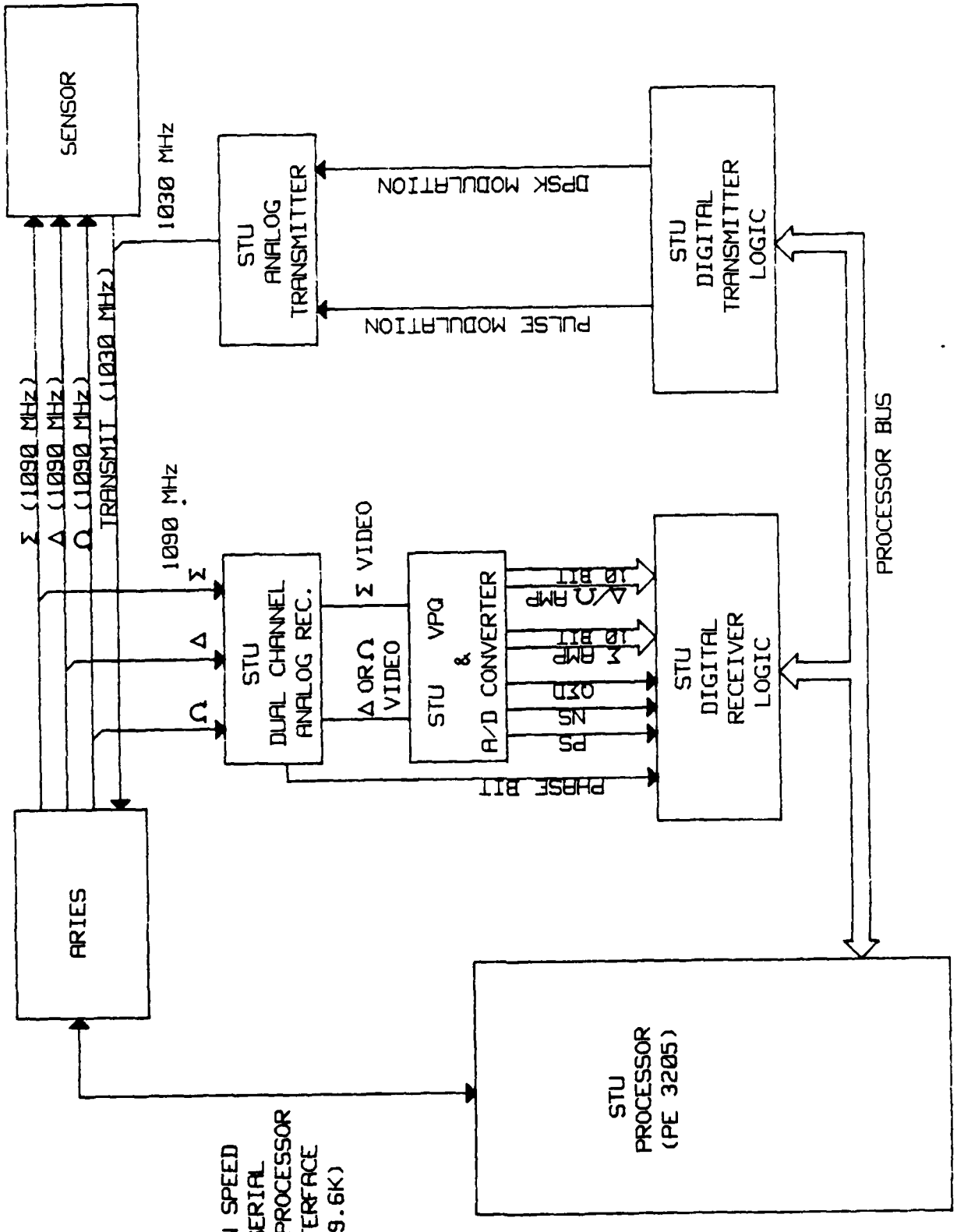


FIGURE 6.10-1. STU/ARIES CONFIGURATION BLOCK DIAGRAM

6.10.1.1 Digital Circuitry.

The digital Uplink Transmitter is designed to simulate the interrogation transmission functions performed by the Mode S sensor. It is capable of handling the same interrogations handled by the Mode S sensor. It supports the transmission of any data pattern in a short or long format including the parity encoding. A functional block diagram of the STU Transmitter is shown in figure 6.10.1.1-1. The individual modules of the transmitter will be discussed in detail in the following subsections.

6.10.1.1.1 Input Buffer and Load Control Logic.

A block diagram of the Input Buffer and the Load Control logic is shown in figure 6.10.1.1.1-1. The Input Buffer consists of a 16-bit x 16-word FIFO and a 16-bit parallel-to-serial converter. The FIFO, used to hold the data (message and address) for one Mode S interrogation, passes the data to the 16-bit parallel-to-serial converter for serial transmission.

Before an interrogation block can be loaded into the transmitter, a command must be issued by the computer specifying such a load will take place. When the command is received, the load enable logic is activated and the load sequence logic is initialized to direct the first word of the interrogation block to its proper destination.

When the first word is sent, the data available instruction DAG1 is fed through a leading-edge detecter to produce a one-clock wide pulse synchronized to the logic clock to strobe the load sequence logic. This produces an active low pulse on the LD RNG1 control line which is used to load the 4-bit interrogation type field into the Type Register and the upper four time-of-transmission bits into the time-of-transmission register discussed in section 6.10.1.1.2. The upper 2 bits of the interrogation type field are fed to the block size decoder via the Type Register and are used to determine the size of the interrogation block to be loaded.

The second data available instruction produces an active low pulse on the LD RNG2 control line which is used to load the second word into the lower 16-bits of the time-of-transmission register. At this time if the interrogation type was ATCRBS (IT0 = 0), the load enable logic would be disabled by the block size decoder preventing any further data loads. When the load enable logic is disabled (meaning that a complete interrogation data block has been loaded), the change in state is detected causing the transmission ready flip-flop IC_1 to be set.

If the interrogation type is Mode S, the load enable logic remains active and all remaining data loads are directed by the load sequence logic to the Input Buffer. After four (seven) additional loads for a short (long) interrogation, the block size decoder disables the load enable logic and the transmission ready flip-flop is set.

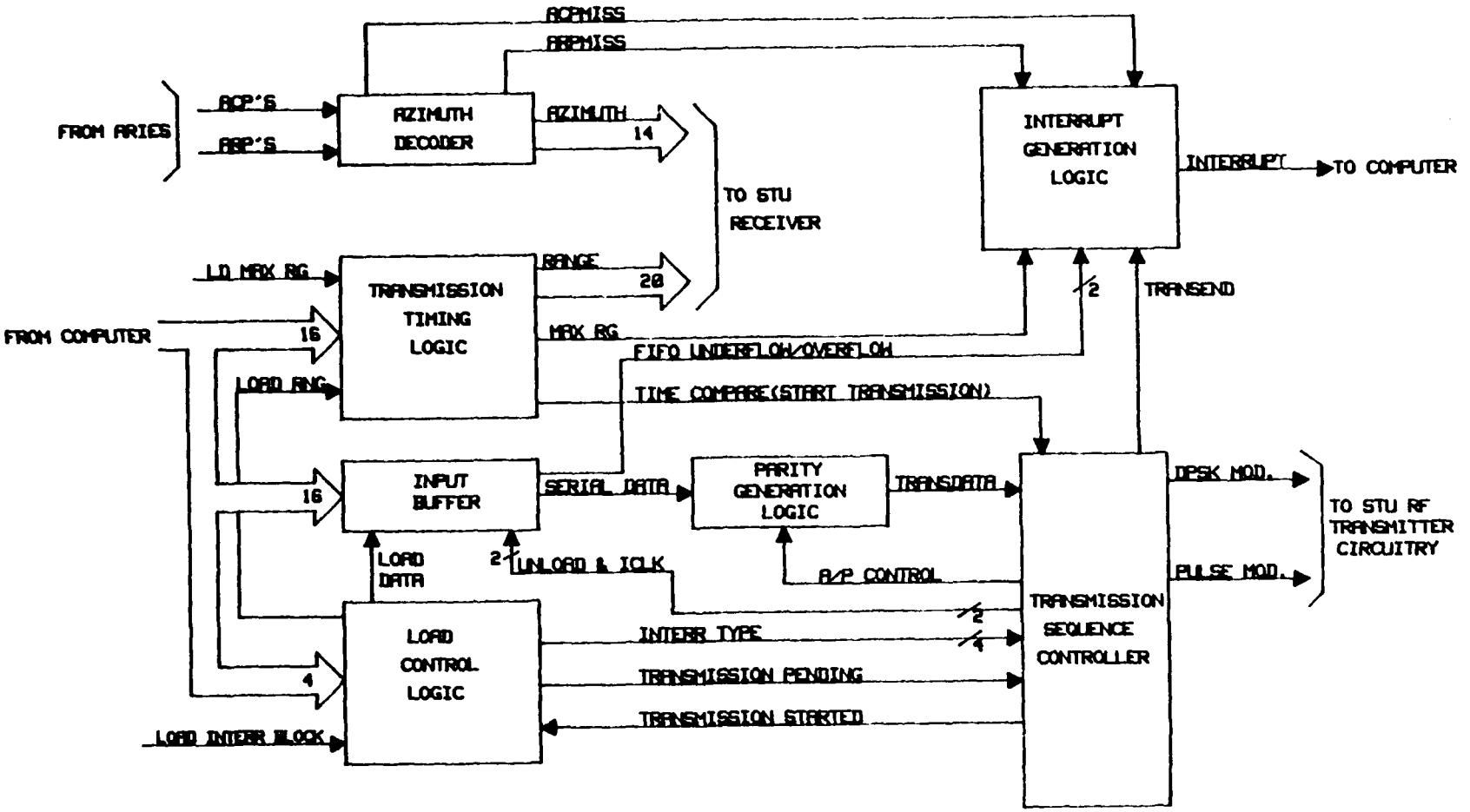


FIGURE 6.10.1.1-1. STU DIGITAL TRANSMITTER FUNCTIONAL BLOCK DIAGRAM

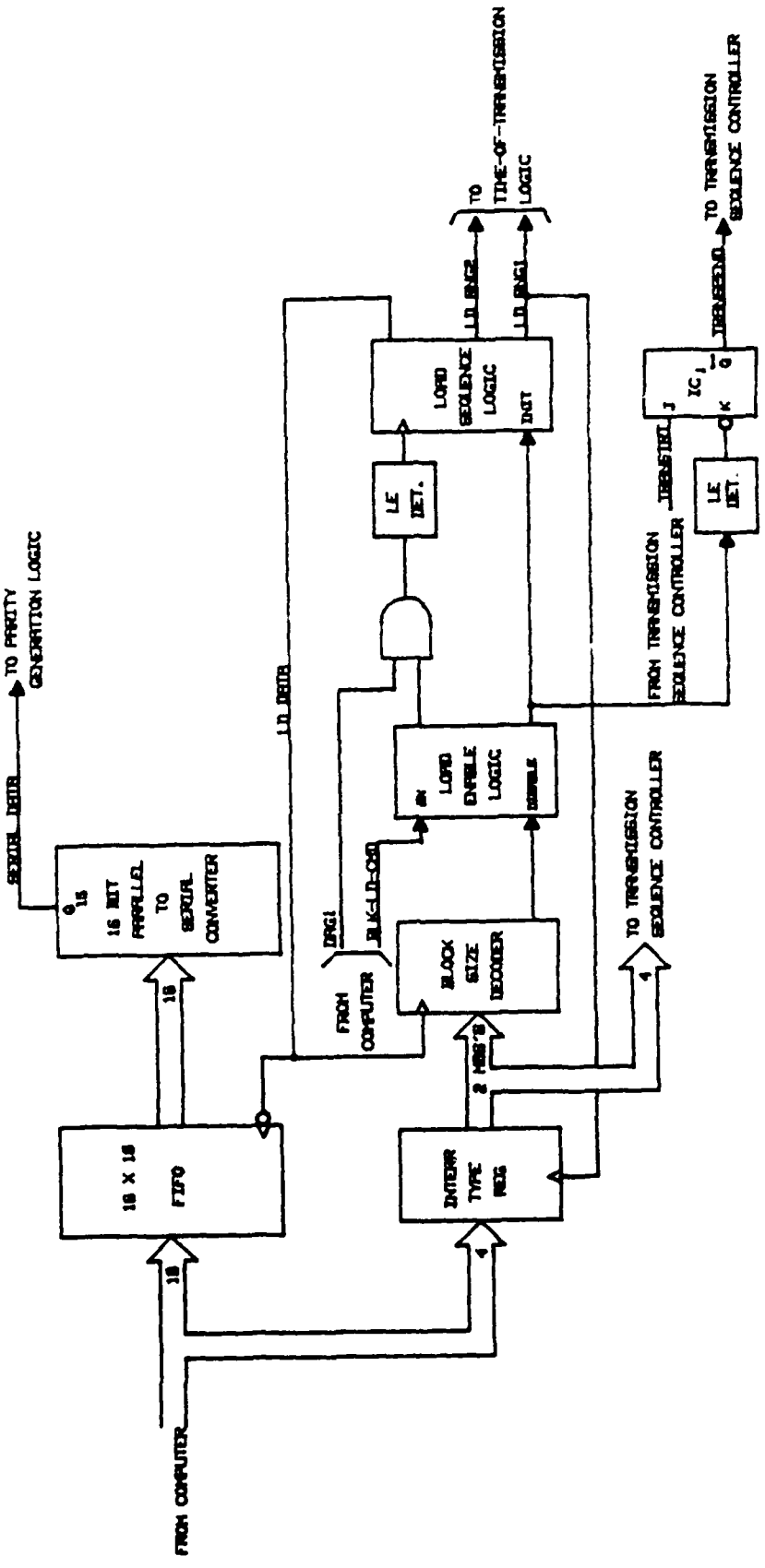


FIGURE 6.10.1.1.1-1. INPUT BUFFER AND LOAD CONTROL LOGIC BLOCK DIAGRAM

6.10.1.1.2 Transmission Timing Logic.

Figure 6.10.1.1.2-1 shows a block diagram of the Transmission Timing logic. This logic consists of a maximum range limit register REG_1, a time-into-transmission-period counter CNTR, an interrogation transmission time register REG_2, a transmission period comparator $COMP_1$, a transmission match comparator $COMP_2$ along with additional support logic. Register REG_1 is a 16-bit register constructed from two octal D-type registers. Register REG_2 is a 24-bit register constructed from three octal D-type registers. Twenty bits of REG_2 hold the time-of-transmission for an interrogation. Counter CNTR is a 20-bit counter constructed from five 4-bit binary counters. Comparator $COMP_1$ is a 16-bit comparator constructed from two 8-bit magnitude comparators, and comparator $COMP_2$ is a 20-bit comparator constructed from three 8-bit magnitude comparators.

In the normal mode of operations, the maximum range limit register is loaded during initialization with the All-Call-to-All-Call interrogation period. The output of this register is placed on the P-inputs of comparator $COMP_1$. The upper 16 output bits of CNTR are placed on the Q-inputs of $COMP_1$. When interrupt generation is no longer disarmed, CNTR is placed in the run state and increments starting at 0 at the rate of the 16-MHz clock. When the upper 16 bits of CNTR matches the value held in REG_1, $COMP_1$ generates the pulse /MAXRNGINT which is used to set CNTR back to 0 and to initiate a maximum range interrupt request on the following clock cycle. This period is then repeated.

A timing mark is generally taken from the maximum range interrupt to load the All-Call interrogation into the Uplink Transmitter. The first two words of the interrogation block are loaded into REG_2. These two words hold the time-of-transmission of the interrogation. The output of REG_2 is placed on the P-inputs of $COMP_2$. The output of CNTR is placed on the Q-inputs of $COMP_2$. When the interrogation time-of-transmission matches the time-into-transmission period, $COMP_2$ generates the pulse TRANSCOMP which is used to trigger the Transmission Sequence Controller (section 6.10.1.1.5) and to initiate an interrogation transmitted interrupt (section 6.10.1.1.6).

Loading Roll-Call interrogations begin after the first interrogation transmitted interrupt is generated in the transmission period. Note, if the transmission time of an interrogation is greater than the maximum range, transmission will not occur. Also, the transmission time of a following interrogation must not occur before the transmission of the previous interrogation is completed. This would also result in no transmission of the following interrogation.

Also, the output of CNTR is sent to the Downlink Receiver via driver array DR made up of 20 inverters.

6.10.1.1.3 Parity Generation Logic.

The Uplink Transmitter is designed with a 24-bit parity encoder. Its purpose is to take the message portion of the interrogation and convert it to 24 parity check bits and to add the remaining address bits with the parity check bits in modulo-two fashion. The 24 parity check bits are computed in the parity encoder register as seen in figure 6.10.1.1.3-1. At the end of the message bits (32 for a Mode S short interrogation and 88 for a Mode S long interrogation), the last 24 bits of the interrogation will be coming directly from the parity encoder rather from the parallel-to-serial shift register through multiplexer MUX_1. Multiplexer MUX_1 is controlled by the Transmission Sequence Controller. (See section 6.10.1.1.5.)

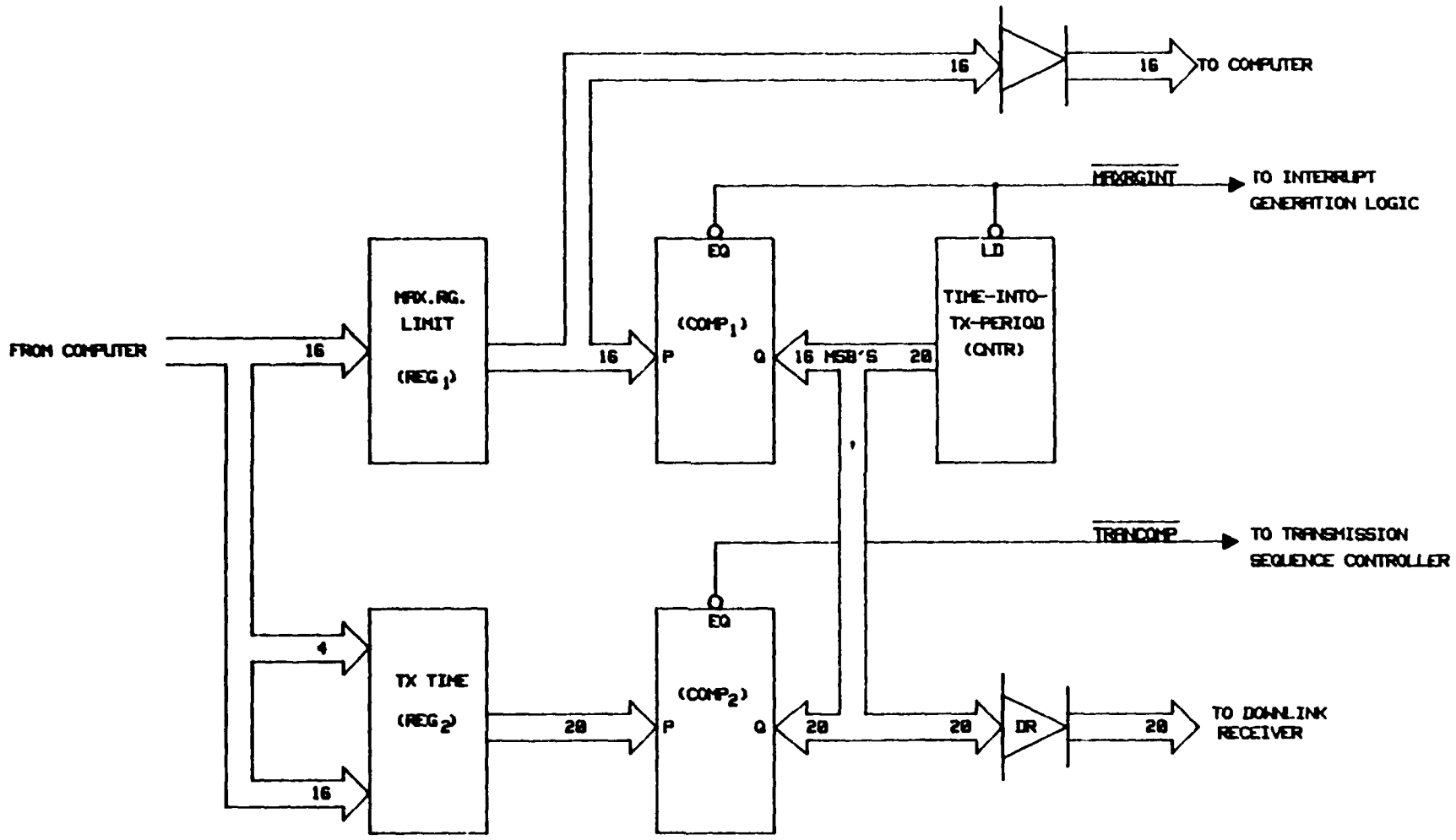


FIGURE 6.10.1.1.2-1. TRANSMISSION TIMING LOGIC BLOCK DIAGRAM

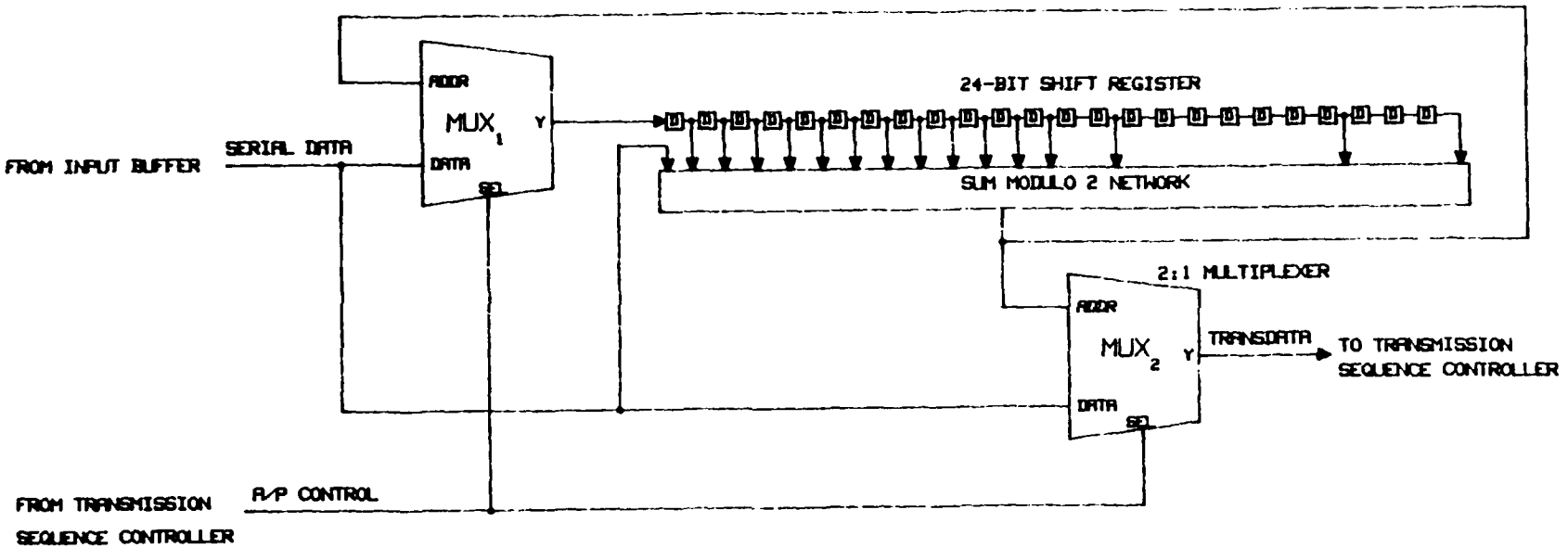


FIGURE 6.10.1.1.3-1. MODE S PARITY ENCODING LOGIC BLOCK DIAGRAM

6.10.1.1.4 Azimuth Decoder Logic.

The Azimuth Decoder block diagram is shown in figure 6.10.1.1.4-1. This logic receives ACPs and ARPs from the ARIES Azimuth Generator/Decoder (section 6.3) which are corrected to true north. The Azimuth Decoder generates 14-bit azimuth words synchronized to the azimuth words used by the ARIES. This same azimuth information is made available to the Downlink Receiver so that received replies can be azimuth tagged to determine if the ARIES transmitted them at the correct azimuth.

Referring to figure 6.10.1.1.4-1, the ACPs and ARPs are lead-edge detected to produce one-clock wide pulses synchronized to the STU's 16-MHz system clock before they are fed to other logic. The ACPs are passed to the input of the 14-bit serial-to-parallel converter CNTR which holds the azimuth word. The output of the 14-bit serial-to-parallel converter is fed to the output register REG, which in turn, is connected to a tristate bus. The computer reads this bus via the GPI of the Uplink Transmitter. Also, the azimuth word is transferred to the Downlink Receiver using the driver array DR.

ACP/ARP error detection logic is provided to insure that the azimuth information is correct. This logic is identical to that of the ARIES Azimuth Generator/Decoder logic and will not be repeated here. (See section 6.3.3, Azimuth Error Detection for further information.)

6.10.1.1.5 Transmission Sequence Controller.

A block diagram of the Transmission Sequence Controller is shown in figure 6.10.1.1.5-1. This controller consists of a ROM address counter which drives a set of three ROMs. Two of the ROMs contain the PAM modulation timing pulses necessary to produce all of the required ATCRBS and Mode S type interrogations. The third ROM is programmed with all of the logic control sequences necessary to transmit any interrogation.

Which transmission sequence is selected is controlled by the type field when it is loaded into register REG_1 at the start of the transmission sequence. The output of this register is fed to the two PAM Modulation ROMs enabling the appropriate ROM and selecting the correct output line via multiplexer MUX_1. Also, this field is used to select the correct transmission end (complete) pulse, which in turn, is used to generate a transmission completed interrupt and if the interrogation is a Mode S type, this field selects the appropriate address/parity control line that is sent to the parity generation logic.

6.10.1.1.6 Interrupt Generation Logic.

The Uplink Transmitter can generate an interrupt for six different requests. In order for the computer to respond to the appropriate request, the interrupt type must be known. This is accomplished through a 3-bit field which is part of the device status byte read automatically by the computer each time an interrupt is acknowledged. The 3-bit field contains the encoded value representing the type of interrupt requested. Table 6.10.1.1.6-1 lists the types of interrupts possible.

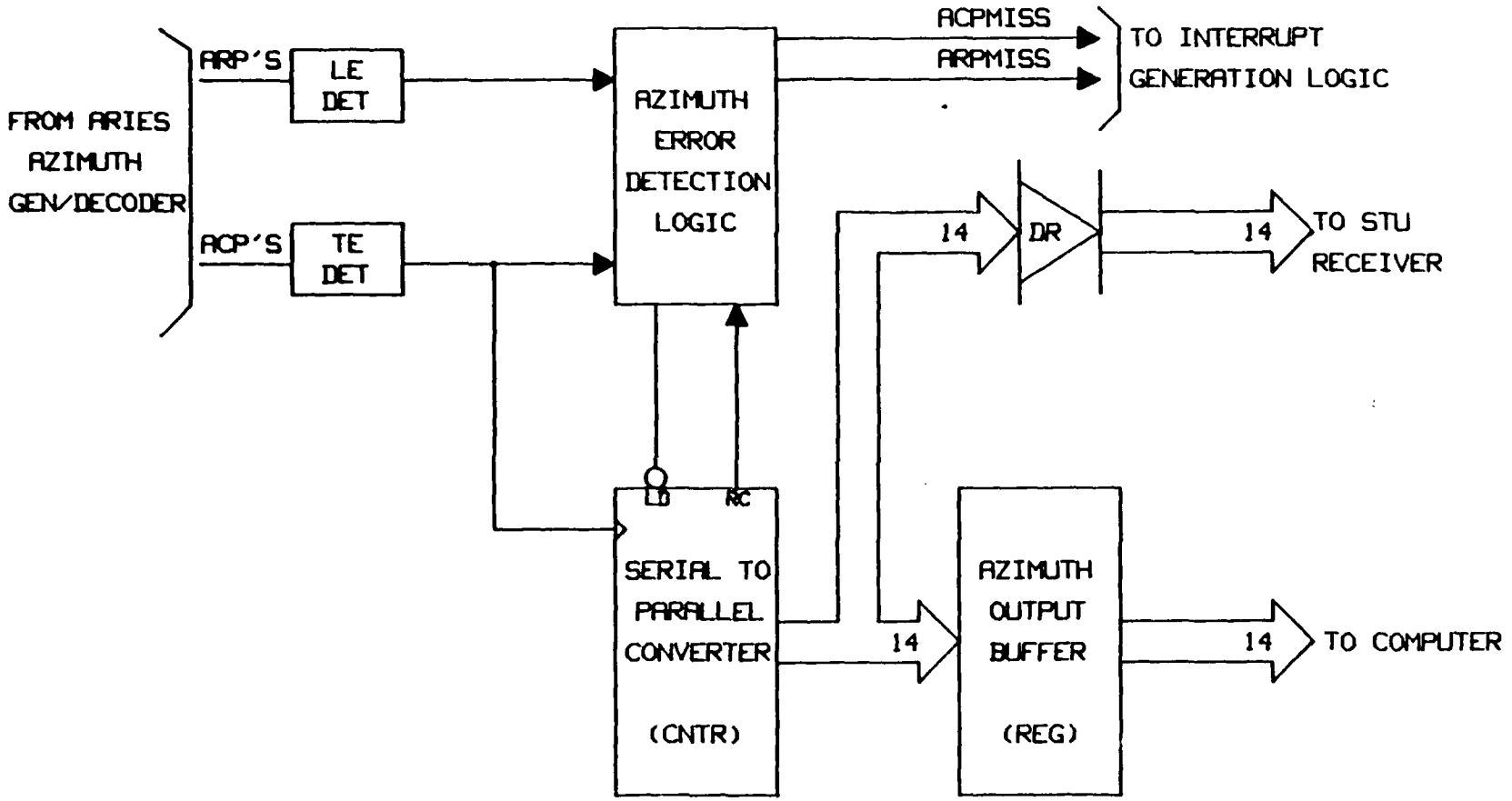


FIGURE 6.10.1.1.4-1. AZIMUTH DECODER LOGIC BLOCK DIAGRAM

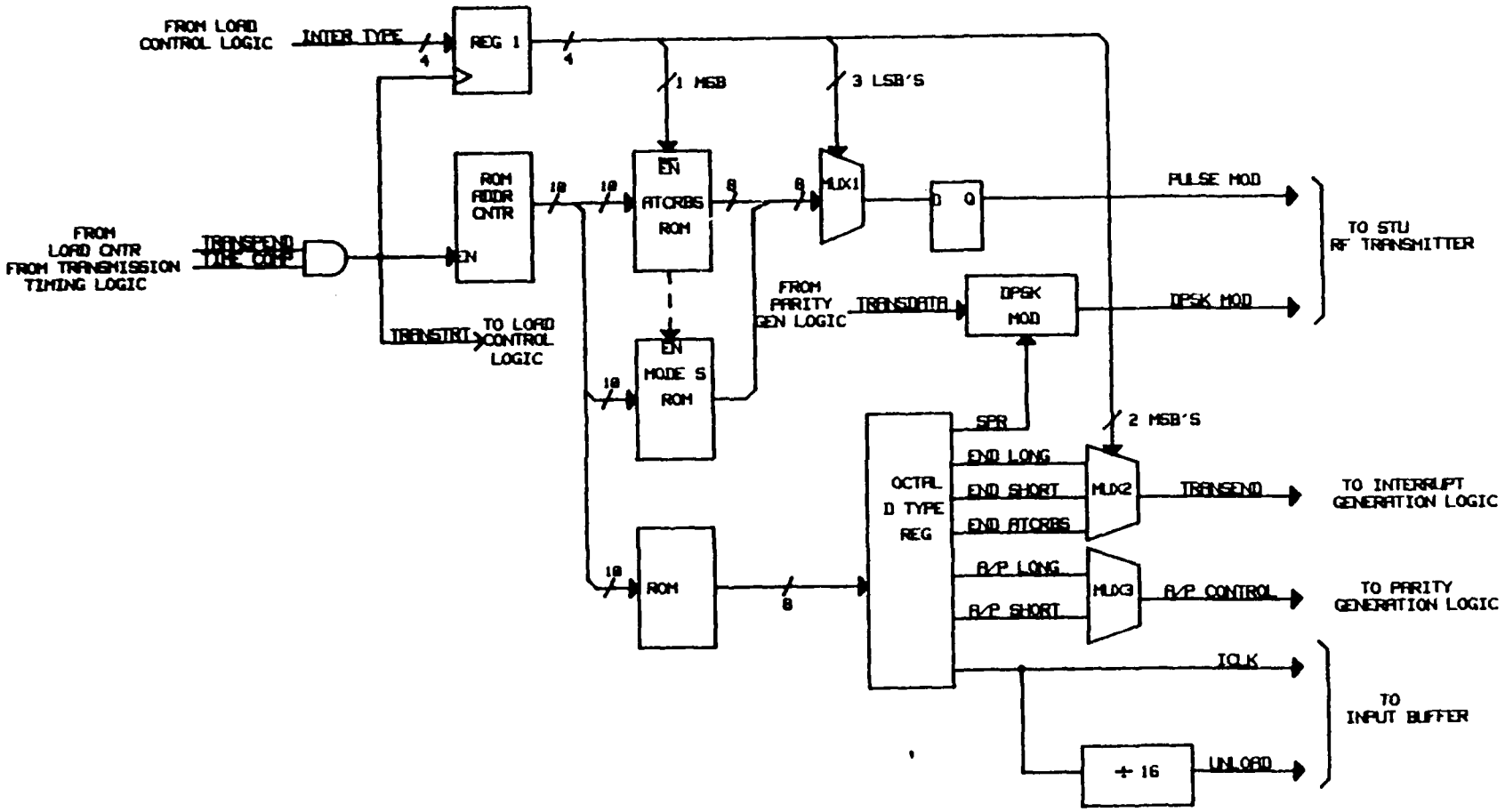


FIGURE 6.10.1.1.5-1. TRANSMISSION SEQUENCE CONTROLLER LOGIC BLOCK DIAGRAM

TABLE 6.10.1.1.6-1. INTERRUPTS GENERATED BY THE UPLINK TRANSMITTER

NMEMONIC	CONDITION	STATUS CODE		
		ST8	ST9	ST10
TRANEND	Interrogation Transmission Completed	0	0	0
MAXRGINT	Start Range Trigger	0	0	1
FIFOFULL	Input Buffer Overflow	0	1	0
FIFOUDRN	Input Buffer Underflow	0	1	1
ACPMISS	Missing ACPs	1	0	0
ARPMISS	Missing ARP	1	0	1

The Interrupt Generation logic is shown in figure 6.10.1.1.6-1. This logic consists of six J/K flip-flops, referred to as interrupt request flip-flops, one for each possible type of interrupt. The /Q output of each flip-flop is fed to the 8-to-3 priority encoder to produce the code specifying the interrupt to be serviced first. Two input lines of the priority encoder are not used and so are tied inactive high. The output of the priority encoder is fed to a register which forms part of the status byte read by the computer. Also, the /Q output of the interrupt request flip-flops are logically OR'ed together and fed to a single shot to produce a pulse which is used to load the output of the priority encoder into the register. This same pulse is fed to the D flip-flop IC_7 to produce the send-attention pulse, SATNO.

The output of the register is fed back to the 3-to-8 line decoder. Each output line of the decoder is fed to an OR gate, which in turn, is fed to one of the interrupt request flip-flops. The active low output line of the decoder enables the OR gate associated with the interrupt request flip-flop that initiated the action.

A trailing-edge detector is used to produce the active low pulse, /INTERESET, once the status byte is read. This pulse is fed to the remaining input of each OR gate, passing through the OR gate enabled by the 3-to-8 line decoder, to reset the interrupt request flip-flop that initiated the request. Also, this pulse is used to reset the single shot for the next interrupt request.

The process of initiating an interrupt is identical for all six interrupts. Also, handling pending interrupts on a priority bases is handled in a similar manner. To understand the operations of this logic, consider the following sequence when interrogation transmission is completed at the same time the maximum range limit is reached:

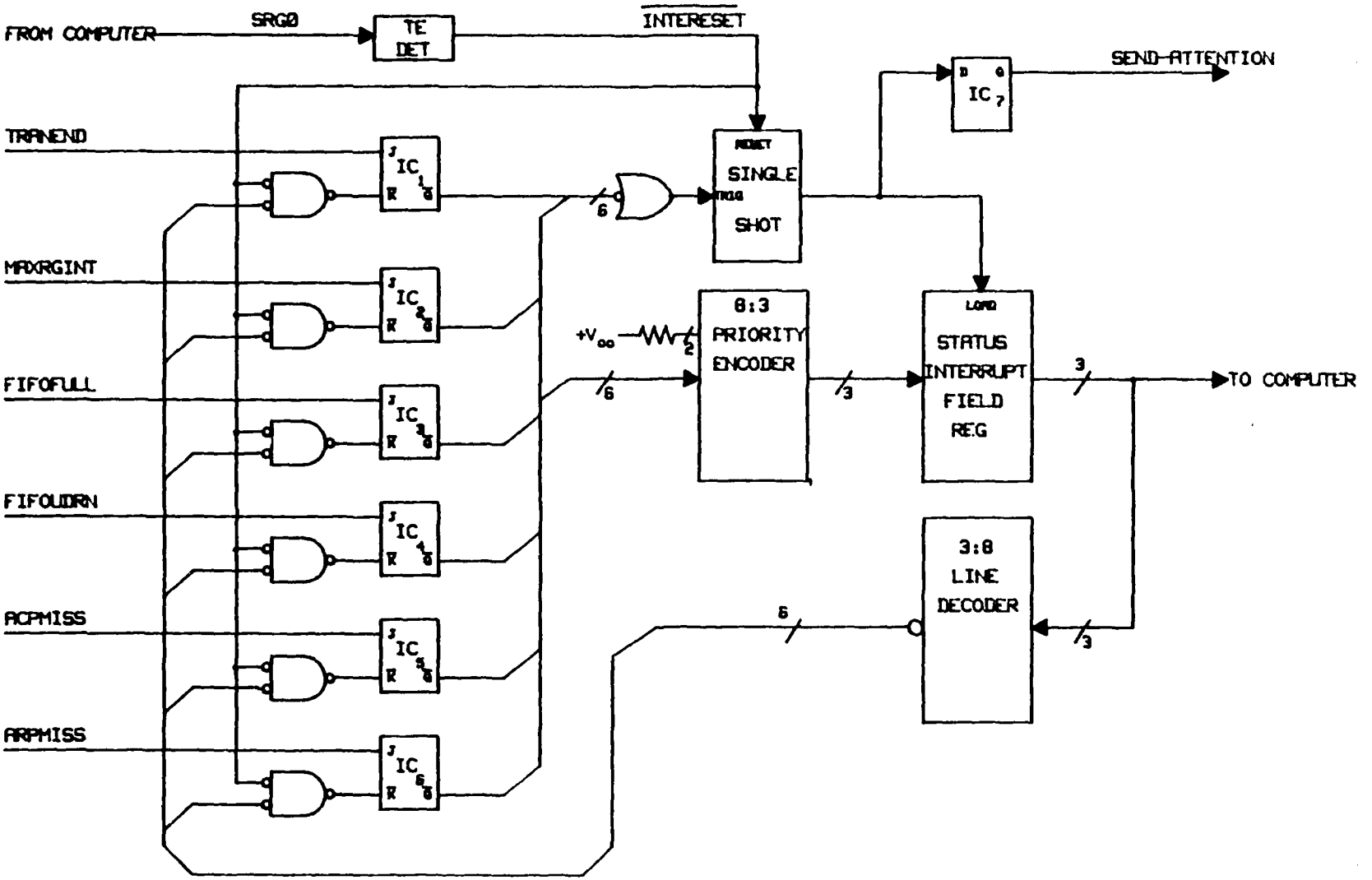


FIGURE 6.10.1.1.6-1. INTERRUPT GENERATION LOGIC BLOCK DIAGRAM

a. Interrupt request pulses TRANEND and MAXRGINT are generated at the same time.

b. J/K flip-flops IC_1 and IC_2 are set on the next clock cycle. With IC_1 set, a low level is applied to the input-7 line of the priority encoder producing the 3-bit output code of "0," independent of the states of the remaining input lines to the priority encoder. At the same time, the register REG is enabled to latch the code.

c. On the following clock cycle, the code is latched into register REG, the single shot is triggered, and the interrupt is generated.

d. After the computer acknowledges the interrupt request and reads the status byte, the active low pulse /INTERESET is placed on the /K input of IC_1 and on the reset input of the single shot.

e. On the next clock cycle, IC_1 is reset along with the single shot. Since IC_2 remains set, the priority encoder changes to a code of 1 and again the register REG is enabled to latch the new code.

f. On the following clock cycle, the new code is latched into register REG, the single shot is triggered, and the second interrupt is generated.

g. After the computer acknowledges the interrupt request and reads the status byte, an active low pulse is passed to the /K input of IC_2 and to the reset input of the single shot.

h. On the next clock cycle, IC_2 is reset along with the single shot completing the interrupt handling sequences.

6.10.1.1.7 Diagnostic Support Logic.

Diagnostic support logic was added to the Uplink Transmitter to provide the capability of verifying its operations as a stand-alone device. The azimuth word that is generated by the Azimuth Decoder logic can be latched in a buffer and read by the computer. The value loaded into the maximum range limit counter and the encoded data specifing the type of interrogation last transmitted can be read by the computer. The data contents of the last Mode S interrogation transmitted can be read, as well.

To encode the type of interrogation transmitted, the PAM control line which is used to drive the analog transmission circuitry is leading and trailing edge detected. The leading edge and trailing edge pulses are then measured to determine the spacing between the PAM pulses and the widths of the pulses. This is accomplished using the logic shown in figure 6.10.1.1.7-1.

The type encoding logic is activated when the first leading edge pulse of an interrogation is detected. This pulse triggers a single-shot circuit on the following clock cycle, starting the ROM address counter at 0. At address 0, the PAM decoder ROM produces the control pulse, CLR-ENCODER1, which resets all of the encoder flip-flops and initializes the interrogation data sampling logic to be covered later.

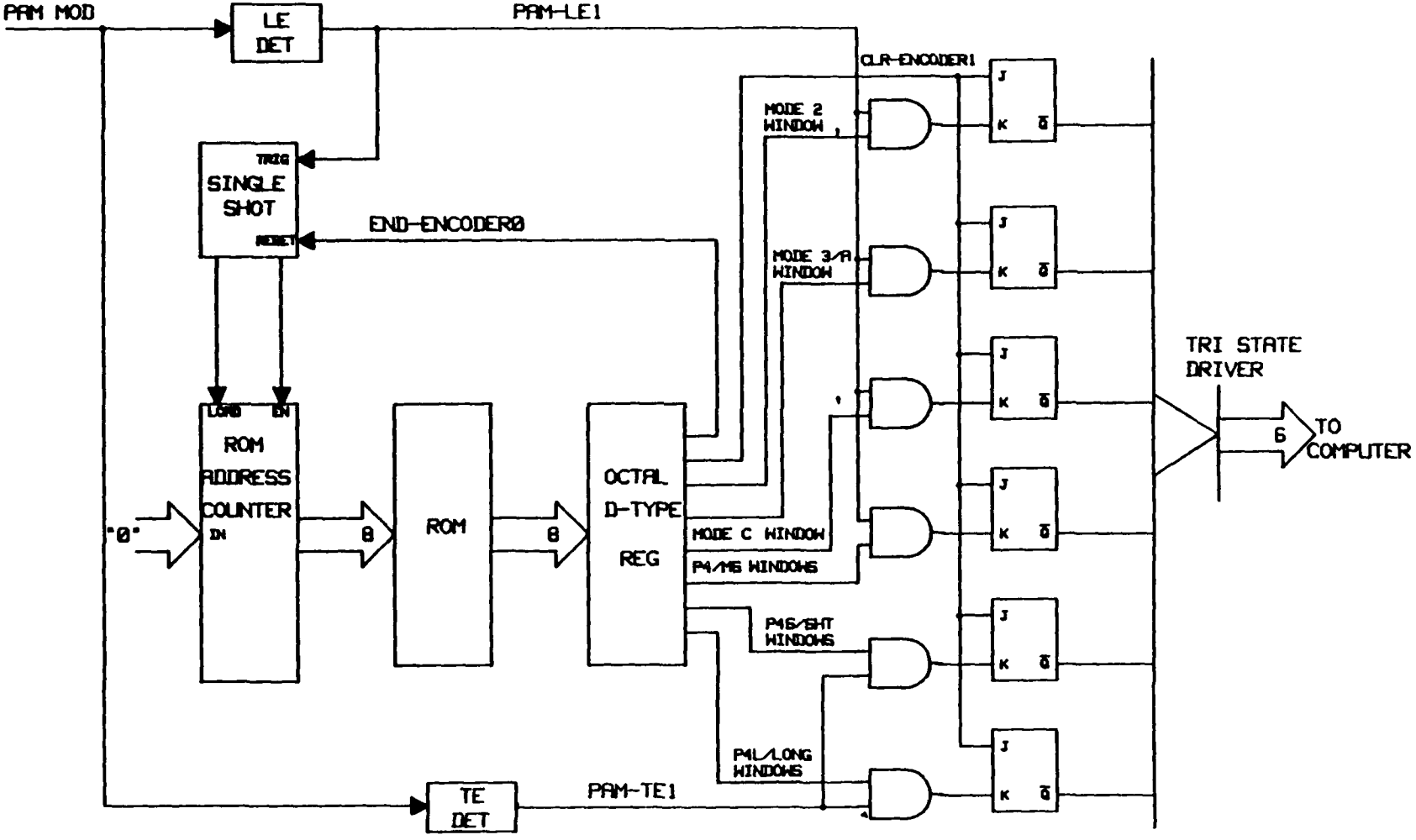


FIGURE 6.10.1.1.7-1. TYPE ENCODING LOGIC BLOCK DIAGRAM

As the address counter continues to increment, the ROM produces a series of timing windows which corresponds to the pulse intervals for ATCRBS mode 2, 3/A, and C brackets, the lead edge of the P4 pulse for each of the ATCRBS brackets, and for the Mode S preamble. To measure the pulse width of the P4 pulse, two trailing edge windows are produced; one corresponding to the pulse width of 0.8 μs and the other corresponding to the pulse width of 1.6 μs. To determine whether a Mode S interrogation is short or long, two P6 trailing edge windows are produced; one corresponding to the 16.25 μs pulse width and the other corresponding to the 32.25 μs pulse width.

Whenever a leading edge or trailing edge pulse occurs within one of the timing windows, the appropriate encoder flip-flop is set. The encoded data for all of the interrogation types is shown in table 6.10.1.1.7-1. Once all of the timing windows have been produced, the ROM produces the control pulse, END-ENCODER0. This pulse resets the single-shot circuit on the following clock cycle, stopping the ROM address counter.

TABLE 6.10.1.1.7-1. ENCODED TYPE INTERROGATION TABLE

INTERROGATION	ENCODER FLIP-FLOPS								ENCODE TYPE FIELD
	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}	
MODE 2	0	0	1	0	0	0	0	0	20
MODE A	0	0	0	1	0	0	0	0	10
MODE A/ONLY	0	0	0	1	0	1	1	0	16
MODE A/ALL-CALL	0	0	0	1	0	1	0	1	15
MODE C	0	0	0	0	1	0	0	0	08
MODE C/ONLY	0	0	0	0	1	1	1	0	0E
MODE C/ALL-CALL	0	0	0	0	1	1	0	1	0D
MODE S SHORT	0	0	0	0	0	1	1	0	06
MODE S LONG	0	0	0	0	0	1	0	1	05

To verify the correct transmission of the Mode S data, the serial data stream TRANSDATA is sampled as it is driving the DPSK modulation flip-flop. This data stream is shifted into a 16-bit serial-to-paralell register. When 16 bits are shifted into the register, it is transferred to the output buffer (FIFO) for storage. The data remains in the output buffer until it is read by the computer or until another interrogation is transmitted. In this case, the CLR-ENCODER1 control pulse, produced by the interrogation type encoder logic, is used to clear the output buffer in preparation for new data.

6.10.1.2 Analog Circuitry.

The analog circuitry for the Uplink Transmitter is capable of supporting all of the Mode S and ATCRBS interrogations as specified in the Mode S National Standard. A block diagram of the analog circuitry is shown in figure 6.10.1.2-1. The modulation signals (PULSE and DPSK) are received from the Uplink Transmitter digital circuitry.

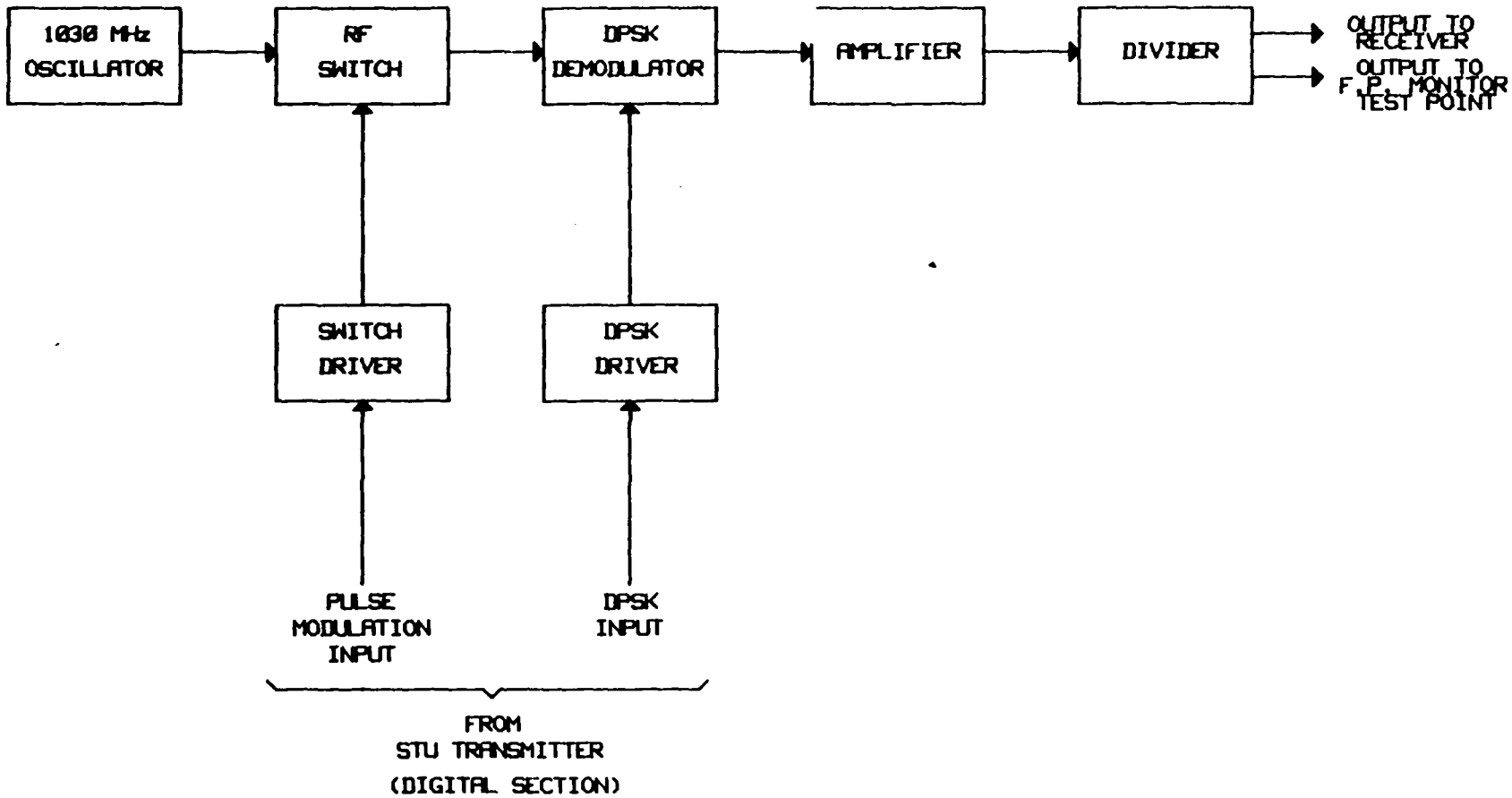


FIGURE 6.10.1.2-1. ANALOG UPLINK TRANSMITTER BLOCK DIAGRAM

The interrogation simulation circuitry contains a crystal-controlled 1030-MHz oscillator which is fed to the input of a RF switch. The RF switch is modulated by the PAM Modulation Driver, which in turn, is driven by the digital PULSE modulation control line received from the digital circuitry. The output of the RF switch is fed to the DPSK Modulator which is a balance mixer type. The 1030-MHz CW message pulse is phase shifted through the DPSK Modulator using the digital DPSK modulation control line received from the digital circuitry. The phase reversal of the RF signal is accomplished by reversing the polarity of the DPSK input to the mixer. The modulated RF signal is then amplified before being split by a power divider. One output is coupled back to the ARIES Uplink Receiver input port and the other to the front panel for monitoring.

6.10.2 Downlink Receiver.

The Downlink Receiver consists of both analog and digital circuitry. The analog circuitry is mounted on the RF to IF Converter panel located in slot No. 11 of the ARIES analog chassis. The video sampling circuitry is mounted on a printed circuit board located in slot No. 1 of the ARIES Digital chassis. A view of the component side of this board is shown in figure 6.10.2-1. The digital receiver circuitry is self-contained on one board located in slot No. 6 of the STU processor chassis.

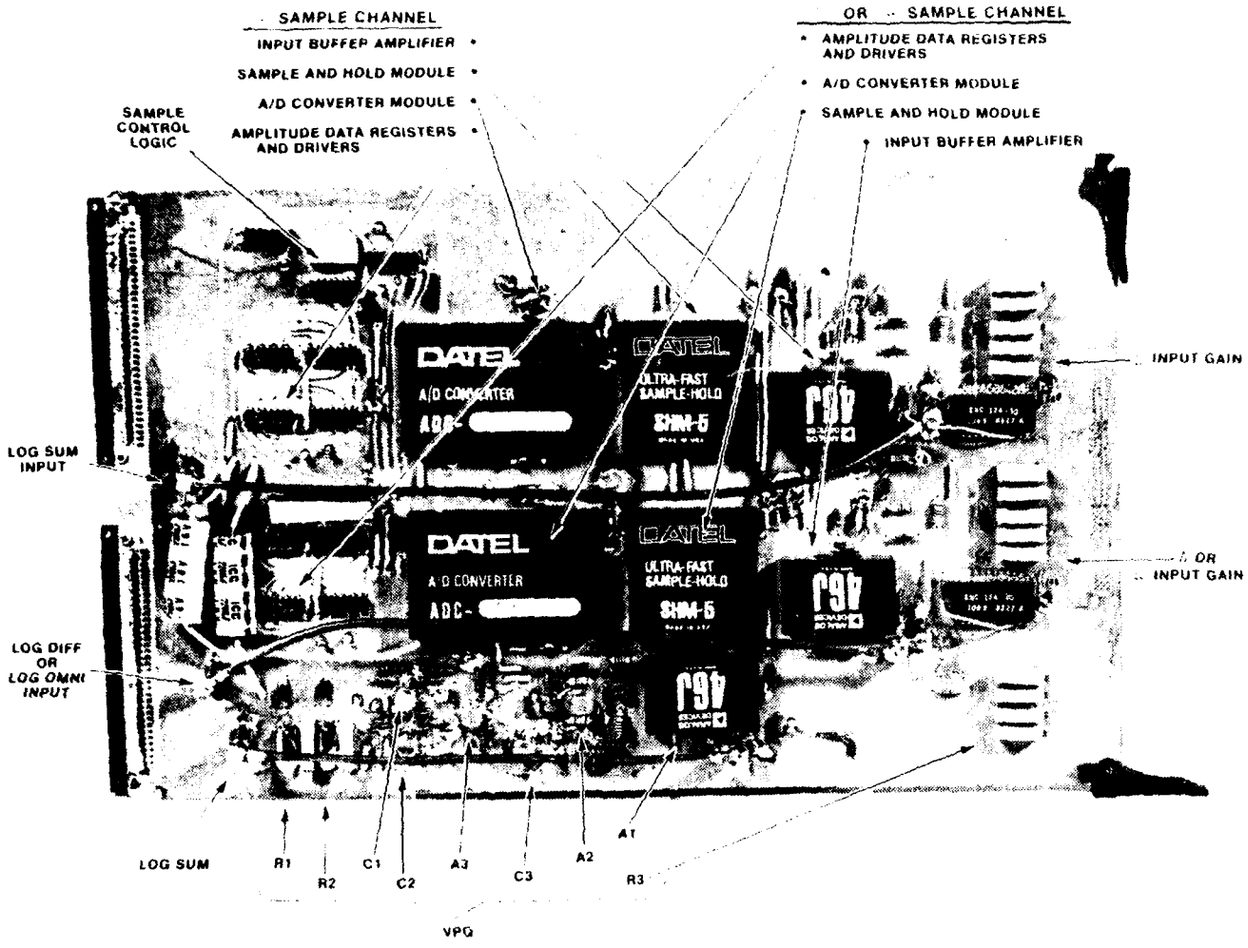
The operations of the analog circuitry are discussed in section 6.10.2.1. The operations of the digital circuitry are discussed in section 6.10.2.2.

6.10.2.1 Analog Circuitry.

The Downlink Receiver samples the SUM, DIFF, and OMNI signals at the 1090-MHz output port of the ARIES. The absolute amplitude of each signal is measured with sufficient accuracy to determine that the level setting circuitry in each reply generator, in particular the monopulse offboresight simulation, is working correctly. The Downlink Receiver measures the relative phase of the SUM and DIFF signals to determine that the monopulse left/right of boresight circuitry is working correctly. Also, the Downlink Receiver samples all reply data bits, for all reply types to determine that serial data transmission and generation of the Mode S reply parity field is working correctly. This is accomplished with the support of the Downlink Receiver analog circuitry.

6.10.2.1.1 RF to IF Downlink Converter.

The RF to IF Converter is shown in figure 6.10.2.1.1-1. This circuitry receives as input the SUM, DIFF, and OMNI signals at 1090 MHz from the IF to RF Converter. (See section 6.4.3.3.) These signals are fed to a three-channel RF Mixer, which is used in conjunction with a 1030-MHz local oscillator, to convert the signals to the 60-MHz IF level. The outputs of the matched three-channel mixer are fed through 60-MHz filters before being fed to 10-dB couplers. The directional couplers are used to direct a small portion of the signal power to test monitoring points located on the front panel. The main output of the couplers are fed to the IF Receiver circuitry.



NOTE: FIGURE 6.10.2.1.3.-1 SHOWS THE LABLED VPQ COMPONENTS IN A SIMPLE CURCUIT DIAGRAM.

FIGURE 6.10.2-1. VPQ COMPONENT SIDE VIEW

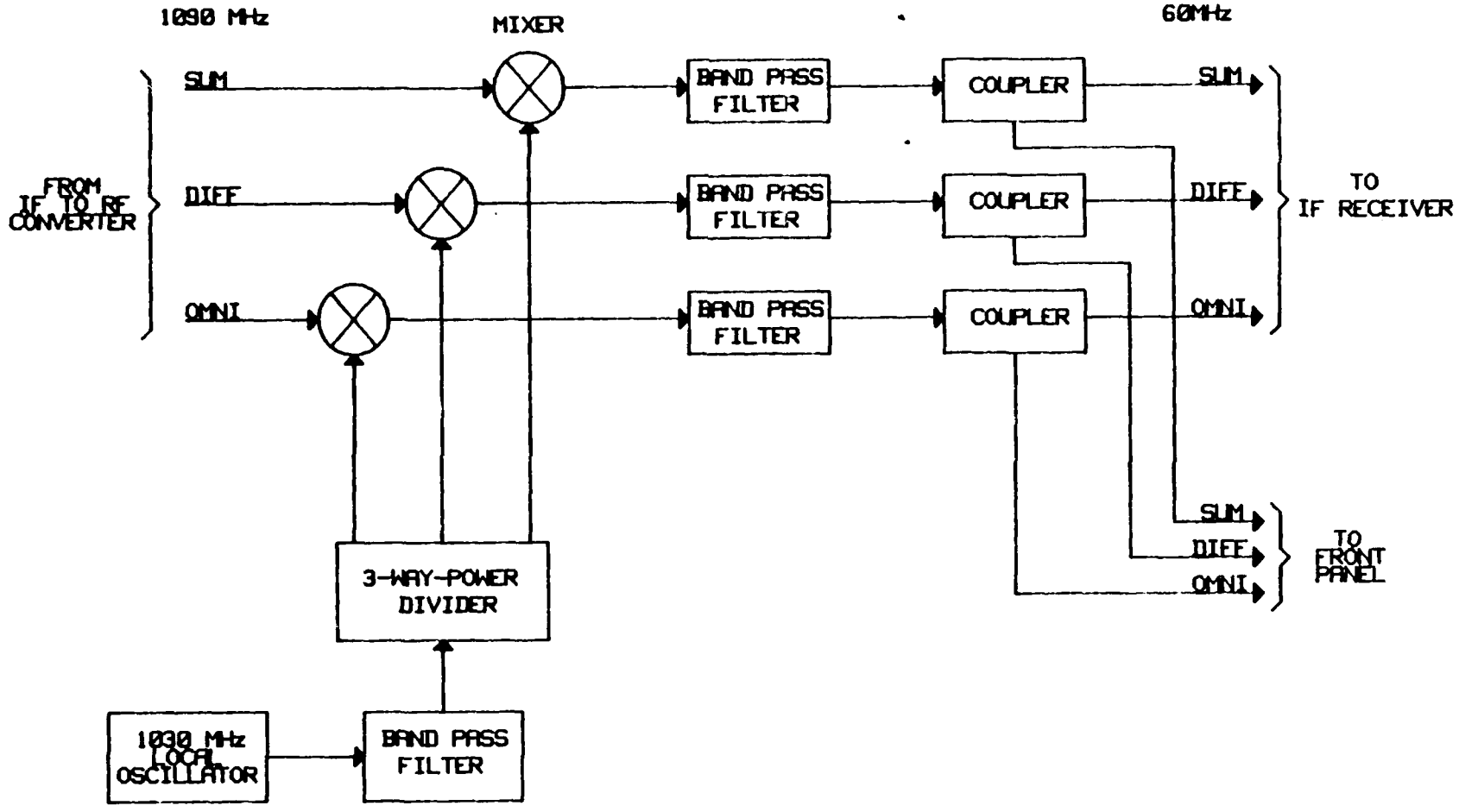


FIGURE 6.10.2.1.1-1. RF TO IF CONVERTER BLOCK DIAGRAM

6.10.2.1.2 IF Receiver.

The IF Receiver circuitry is shown in figure 6.10.2.1.2-1. This circuitry receives as input the SUM, DIFF, and OMNI signals at 60 MHz from the RF to IF Converter covered in the previous section. This circuitry produces as output, three SUM video signals, one DIFF or OMNI video signal, and a relative SUM to DIFF phase measurement signal.

The received SUM signal is power divided into two signals. One of these signals is amplified by a log amplifier before being fed to a three-channel video output driver circuit. Two of the resultant video output signals are sent to the VPQ board where one is quantized by the VPQ circuit for pulse detection and the other is sampled and converted into binary words by separate sample and hold (S/H) and analog to digital converter (A/DC) modules. The third resultant video output signal is connected to the distribution panel for use as a SUM video test point.

The second SUM signal out of the power divider is used for measuring the phase difference (assumed to be 0° and 180°) between the SUM and DIFF signals. This is accomplished by splitting the received DIFF signal using a power divider and multiplying one of these signals with the second SUM signal together in a mixer (phase detector). The result from the mixer is then fed to a phase bit driver circuit to produce the TTL compatable phase bit ("0" for 0° and "1" for 180°). This phase bit provides a means of checking the phase shifter in the reply generator IF Units.

The remaining DIFF signal and the received OMNI signal are connected on the inputs of a SPDT IF switch controlled by the Downlink Receiver digital circuitry. The selected signal is amplified by a log amplifier before being fed to an output driver. The resultant video output signal is sent to the VPQ board where it is sampled and converted into binary words by a second set of S/H and A/DC modules.

6.10.2.1.3 Video Pulse Quantizer.

The input of the VPQ circuit is the LOG SUM signal from the IF Receiver. The VPQ provides the following outputs to the Downlink Receiver video digitizer for pulse detection:

a. Quantized SUM Positive Slope (/PS)
b. Quantized SUM Negative Slope (/NS)
c. Quantized SUM (/QSUM)

These signals are used by the video digitizer to detect and decode replies. The following discussion is a detailed analysis of the individual video pulse quantizer circuits. A simplified block diagram of the VPQ is shown in figure 6.10.2.1.3-1.

The input from the SUM LOG channel is coupled to the linear buffer amplifier A_1. The output of the amplifier is coupled through two paths, one delayed by 135 ns with respect to the other. The subtracting amplifier A_2 subtracts the delayed image of the input pulse from itself, producing a positive and a negative slope of the input signal from the LOG SUM channel. Figure 6.10.2.1.3-2 shows the individual waveforms of the VPQ.

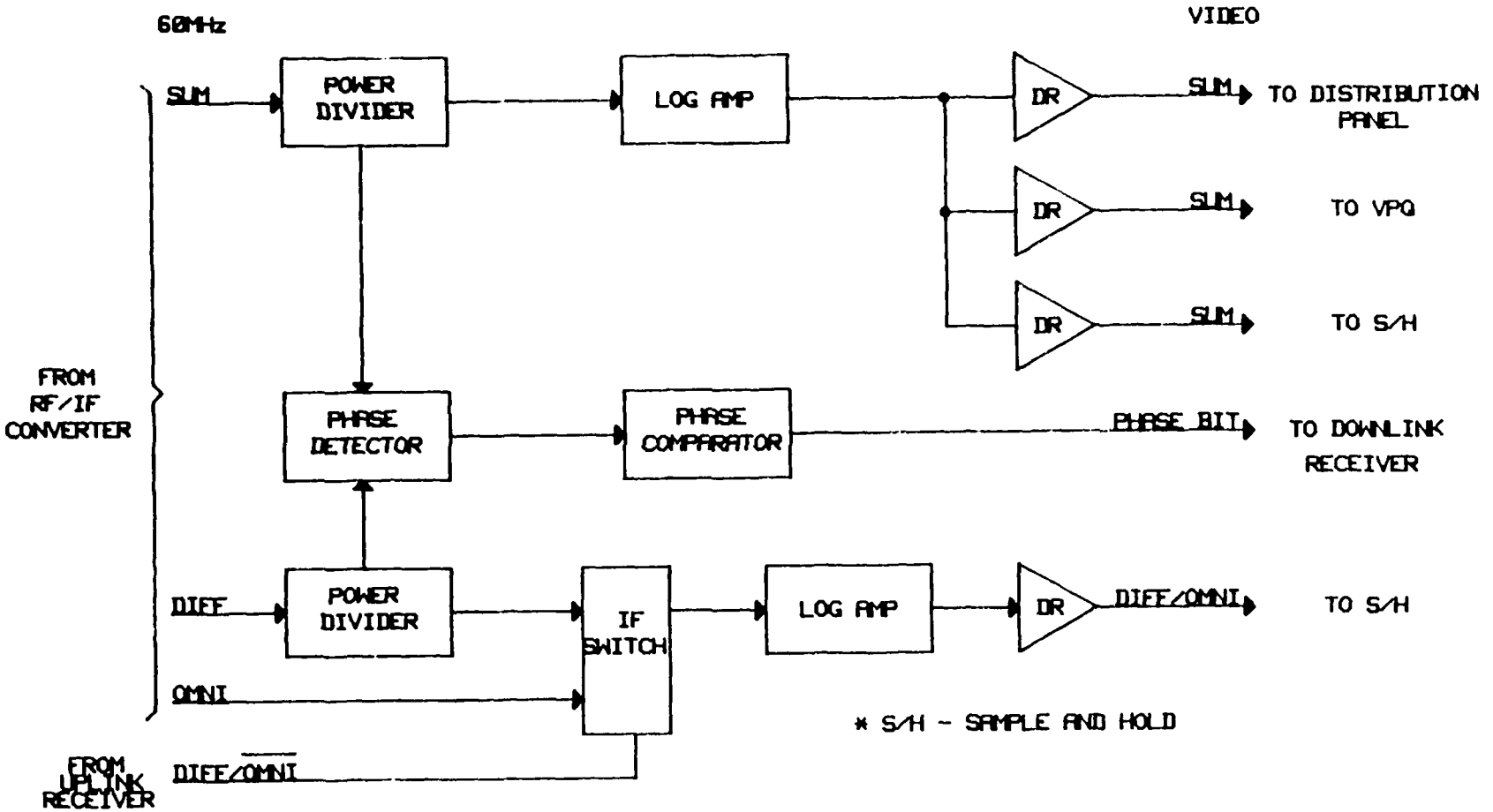


FIGURE 6.10.2.1.2-1. IF RECEIVER BLOCK DIAGRAM

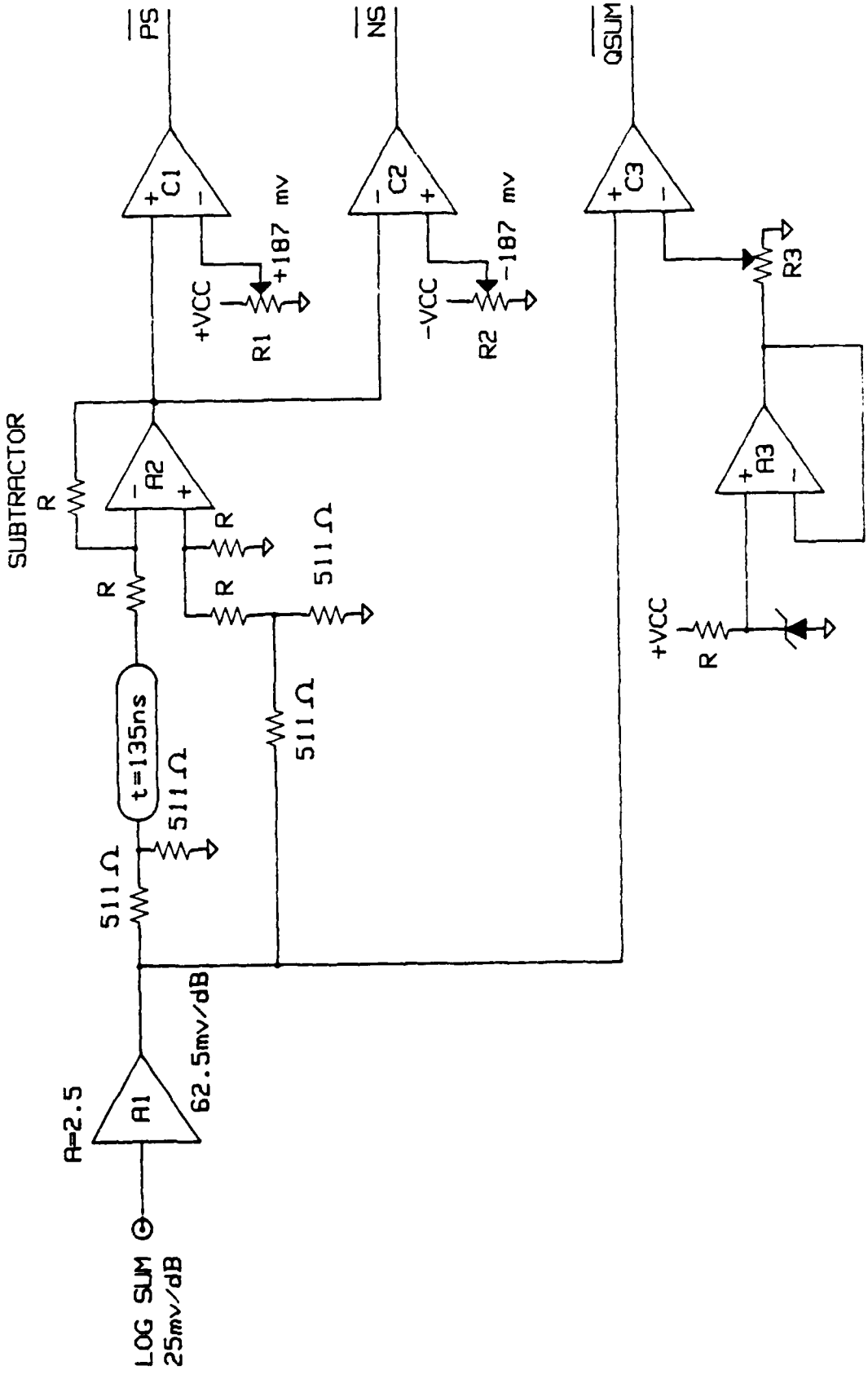


FIGURE 6.10.2.1.3-1. SIMPLIFIED DIAGRAM OF VPQ CIRCUITRY

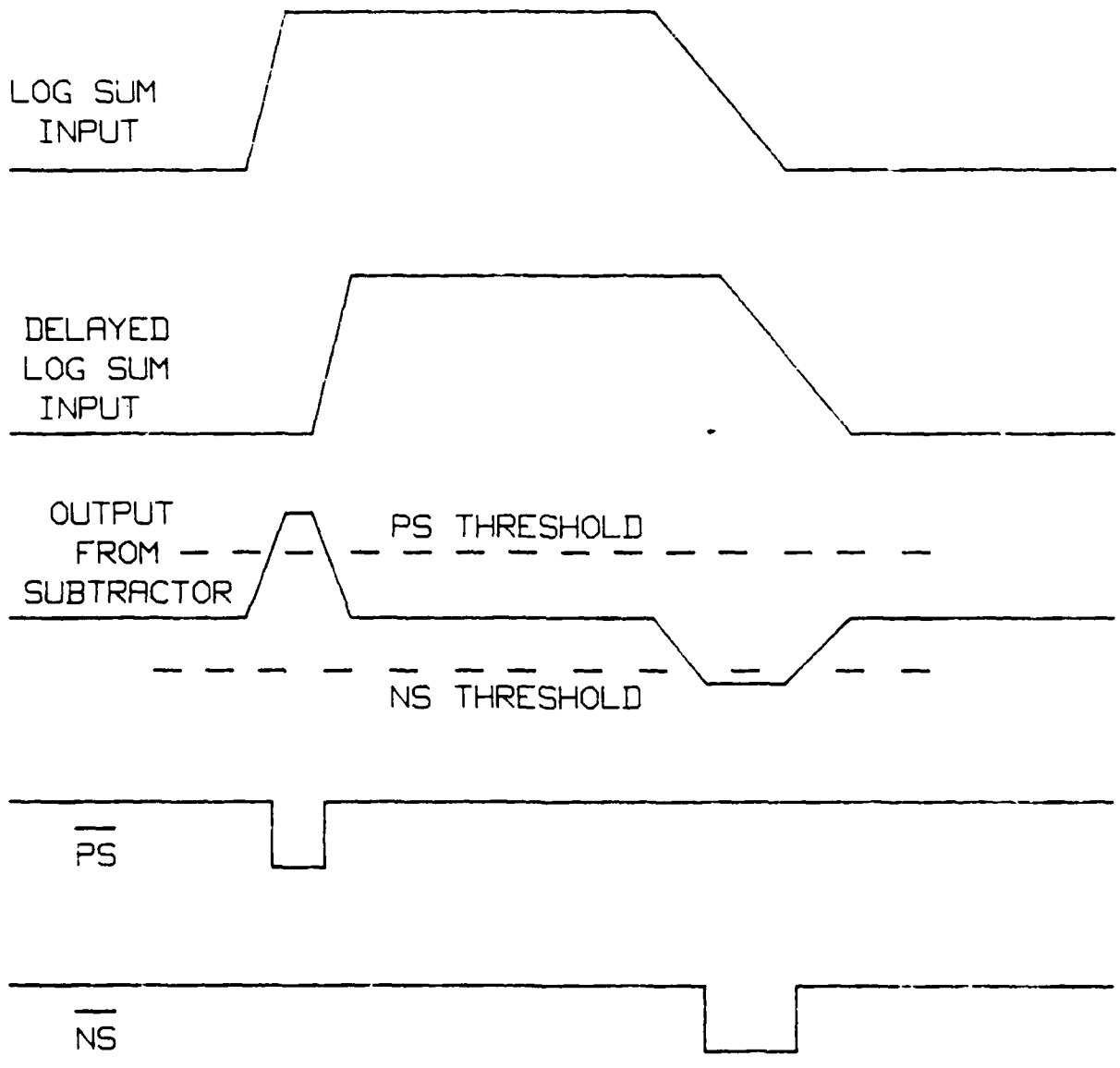


FIGURE 6.10.2.1.3-2. VPQ WAVEFORMS

The output of the subtracting amplifier is applied to two comparators to determine the width of the LOG SUM pulse by defining accurately the location of the leading and trailing edges. The positive slope is detected by comparator C_1 when the pulse exceeds a fixed threshold set by trimming potentiometer R_1. The negative slope is similarly detected by comparator C_2 when the pulse falls below a fixed threshold set by trimming potentiometer R_2.

The output of amplifier A_1 is also coupled to the QSUM comparator C_3 whose bias level is determined by a fixed threshold buffer amplifier A_3. The fixed threshold buffer amplifier applies a voltage to trimming potentiometer R_3 whose wiper determines the fixed threshold. This threshold is normally set such that a signal level of -79 dBm at the Mode S sensor receiver antenna port will exceed the threshold. The QSUM output is "active low" only if the SUM input is above the fixed threshold setting.

6.10.2.1.4 Dual Channel Video Sampling Circuitry.

The video sampling circuitry samples the amplitude of two video channels simultaneously. A simplified block diagram of the dual channel video sampling circuitry is shown in figure 6.10.2.1.4-1. As shown in this figure, one channel is dedicated to sampling the LOG SUM video while the other channel samples either the LOG DIFF or LOG OMNI video. The selection is controlled by the digital Downlink Receiver.

The following discussion is a detailed description of the individual circuits of the LOG SUM video sampling channel. The LOG DIFF/LOG OMNI video sampling channel is identical for that of the LOG SUM and, therefore, will not be repeated.

Before the received LOG SUM is applied to the S/H module, it is further amplified by a linear amplifier providing an input range of 0 to 10 volts. The S/H insures that the input to the A/DC remains constant during conversion time. The A/DC used in the video sampling circuitry is a successive approximation type with a conversion time of 2 μs. When a conversion is finished, an end-of-conversion pulse is generated, causing the result of the conversion to be loaded into an output buffer. When both channel output buffers are loaded, a data ready signal is sent to the digital Downlink Receiver to indicate that new binary amplitude data is available.

The SAMPLE VIDEO control signal issued by the video digitizer logic is coupled to the sample control logic, which in turn, generates the necessary control signals to the S/H and A/DC. The SAMPLE VIDEO control signal is normally in the active low sample mode. When the control signal goes high, the hold strobe to the S/H follows 120 ns later. This is necessary to align the hold strobe with the positive edge of the ATCRBS F_2 pulse. Sixty ns later, the 60-ns-wide start convertion pulse to the A/DC goes active high.

6.10.2.2 Digital Circuitry.

The Downlink Receiver digital circuitry performs similar functions to that of the Mode S sensor receiver. The digital circuitry is capable of decoding all reply types handled by the sensor. It assembles reply data blocks containing similar information, such as, reply data, range (time), boresight azimuth, monopulse data (LOG SUM and LOG DIFF amplitude samples), left/right of boresight detection, etc. It provides the reply data blocks to the STU processor for analysis.

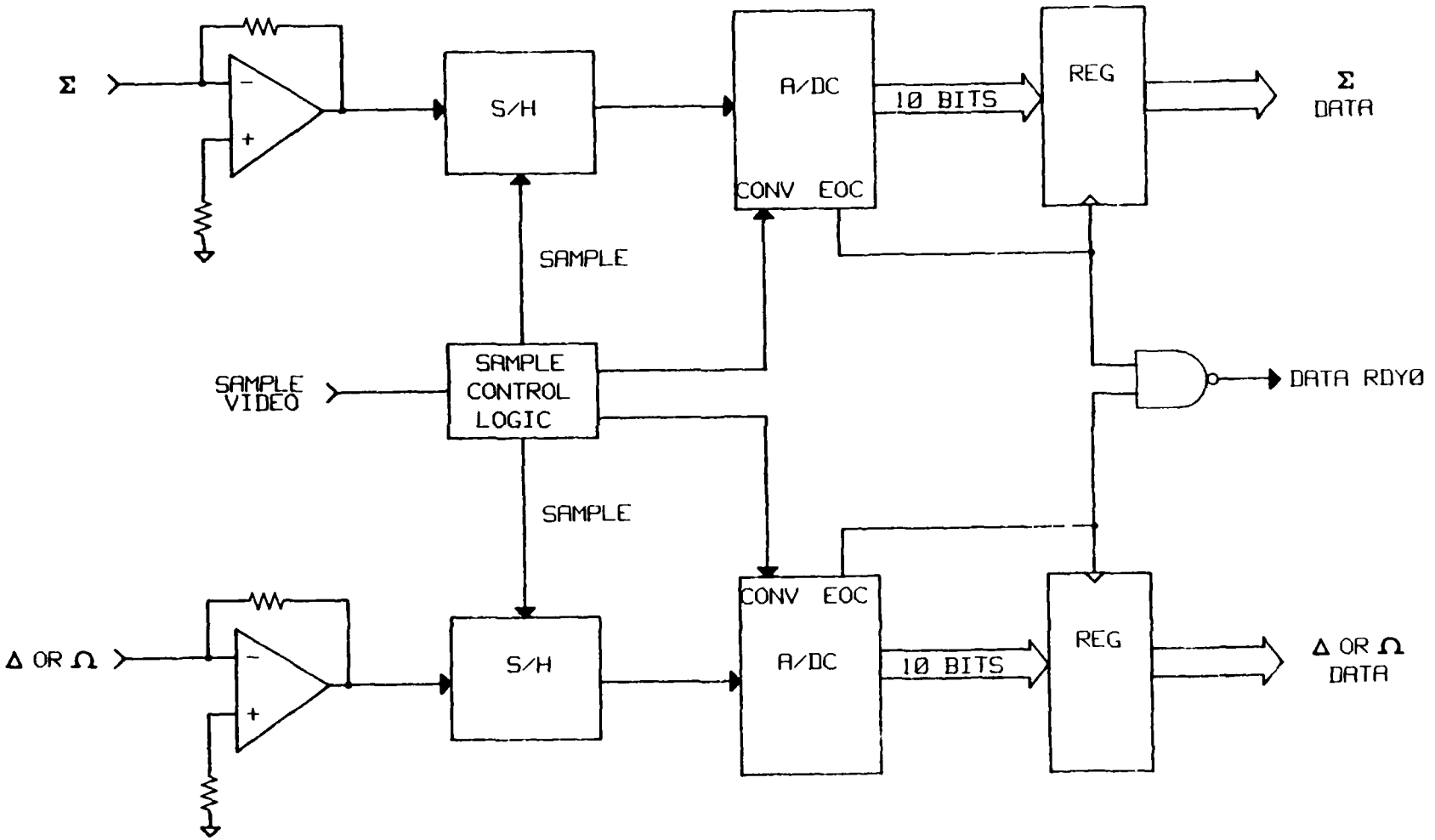


FIGURE 6.10.2.1.4-1. SIMPLIFIED DIAGRAM OF THE DUAL CHANNEL VIDEO SAMPLING CIRCUITRY

A functional block diagram of the Downlink Receiver digital circuitry is shown in figure 6.10.2.2-1. Each block is described in detail in the following subsections.

6.10.2.2.1 STU Logic Clock.

The ARIES and the STU digital logic both use the 16-MHz clock rate but they are not synchronized to one another. The STU digital logic uses its own logic clock independent of that used by the ARIES digital logic. This is done so that the ARIES can be fully tested with an environment asynchronous to itself.

Figure 6.10.2.2.1-1 shows the logic clock and clock distribution used by the Uplink Transmitter and the Downlink Receiver logic. This circuit consists of one 16-MHz crystal controlled oscillator fed to two schottky type inverters. One inverter is used as the driver to provide the logic clock to the digital Uplink Transmitter board. The Transmitter board receives the logic clock by an identical schottky type inverter, which in turn, supplies the clock to a set of high current sink NAND gate buffers for distribution on the Transmitter board. On the Downlink Receiver board, the remaining inverter is fed through a second inverter before being supplied to an identical set of high current sink NAND gate buffers for distribution on the Downlink Receiver board. This is done to maintain the phase alignment of the 16-MHz logic clock on both boards.

6.10.2.2.2 Video Digitizer.

The Video Digitzer shown in figure 6.10.2.2.2-1 accepts as input the /PS, the /NS, and the /QSUM pulses produced by the VPQ circuit. The /QSUM pulse is used for gating the /PS pulse through a leading edge detector (IC_1, IC_2, and AND gate A) to produce a one-clock wide pulse synchronized to the logic clock, which in turn sets the J/K flip-flop IC_5. It remains set until the /NS pulse arrives. The /Q output of IC_5 is used for gating the /NS pulse through a second leading edge detector (IC_3, IC_4, and NAND gate B), resetting flip-flop IC_5. Thus, a TTL pulse with the same pulse width as the input to the VPQ is created.

6.10.2.2.3 Bracket and Preamble Detection Logic.

A block diagram of the bracket and preamble detection logic is shown in figure 6.10.2.2.3-1. The ATCRBS brackets are detected by feeding the TTL pulses generated by the Video Quantizer into a digital delay circuit. The serial output of this logic lags behind the serial input by 20.125 µs. The serial input is leading edge detected and the result is logically AND'ed with the delayed output. When a leading edge pulse occurs and at the same time, a delayed quantized pulse occurs, the bracket detect pulse is generated. The Mode S preambles are detected by feeding the leading edge of the quantized pulses into a 5-µs shift register. The shift register is then tapped at 0, 1, 3.5, and 4.5 µs. The 0, 3.5, and 4.5 µs taps each have tolerances of -62.5/+196.5 ns with the 1-µs tap held fixed. When a leading edge pulse occurs at all four taps simultaneously, the preamble detect pulse is generated. Both pulses are used to start the data sampling logic by setting the Reply in Progress flip-flop.

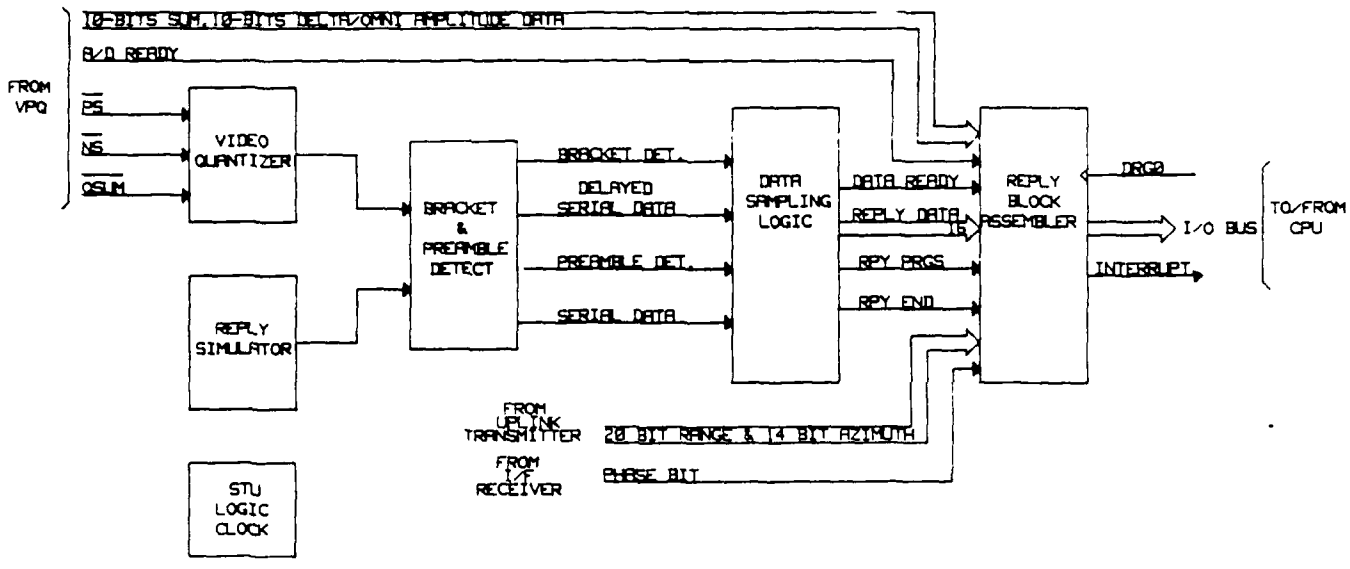


FIGURE 6.10.2.2-1. DIGITAL DOWNLINK RECEIVER BLOCK DIAGRAM

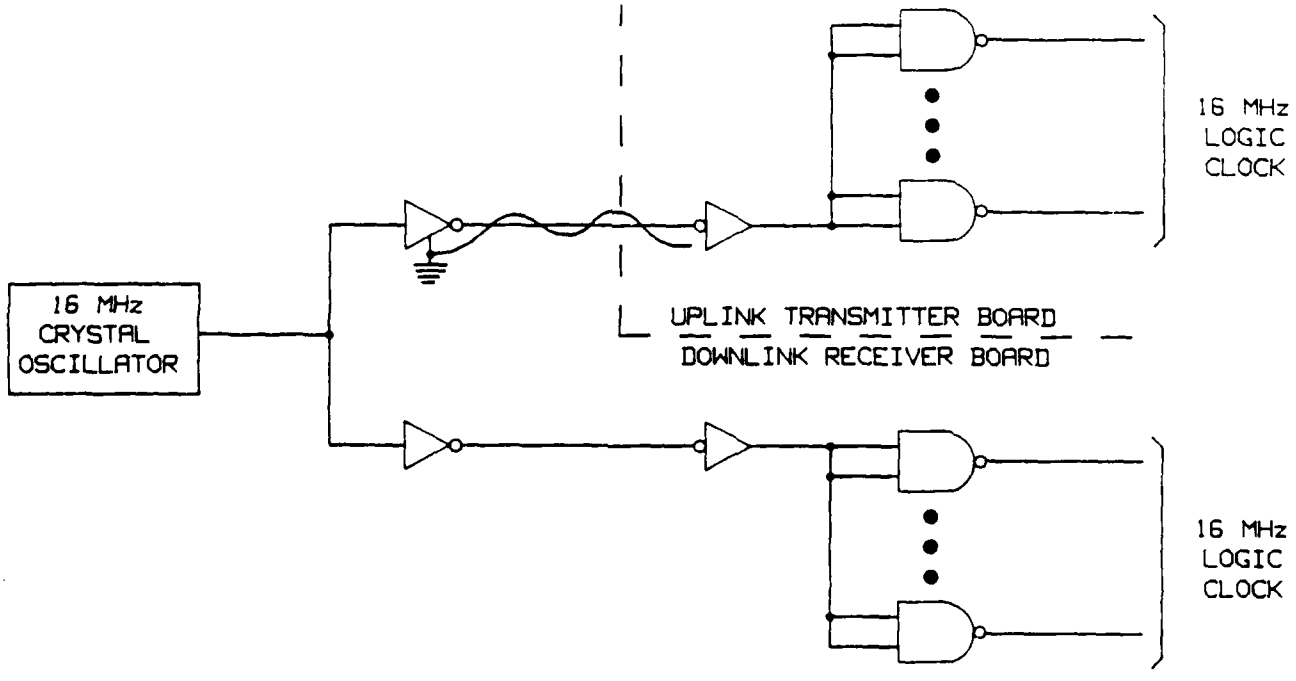


FIGURE 6.10.2.2.1-1. STU CLOCK LOGIC

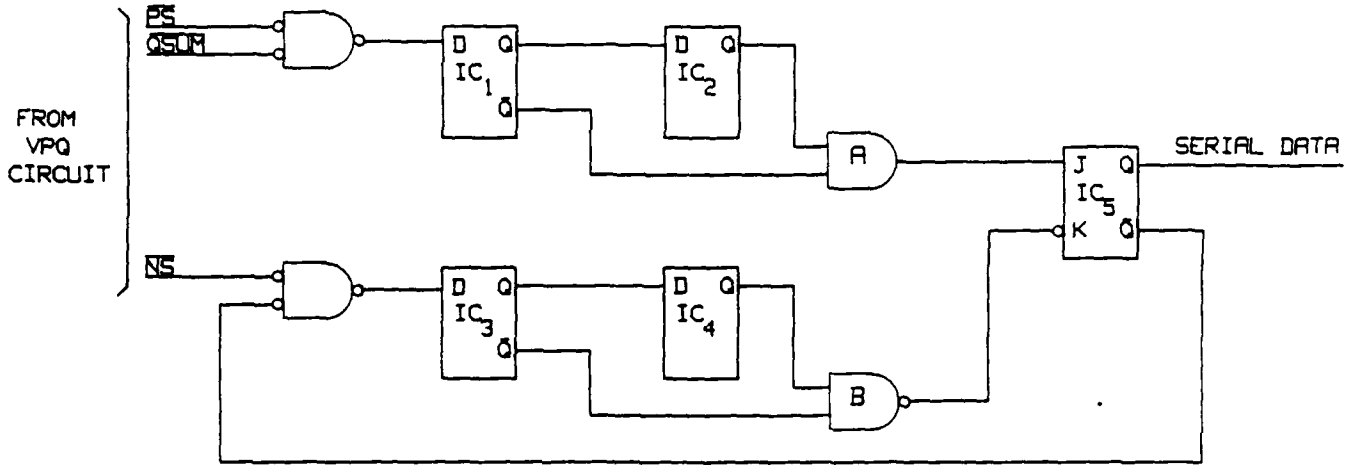


FIGURE 6.10.2.2.2-1. VIDEO DIGITIZER

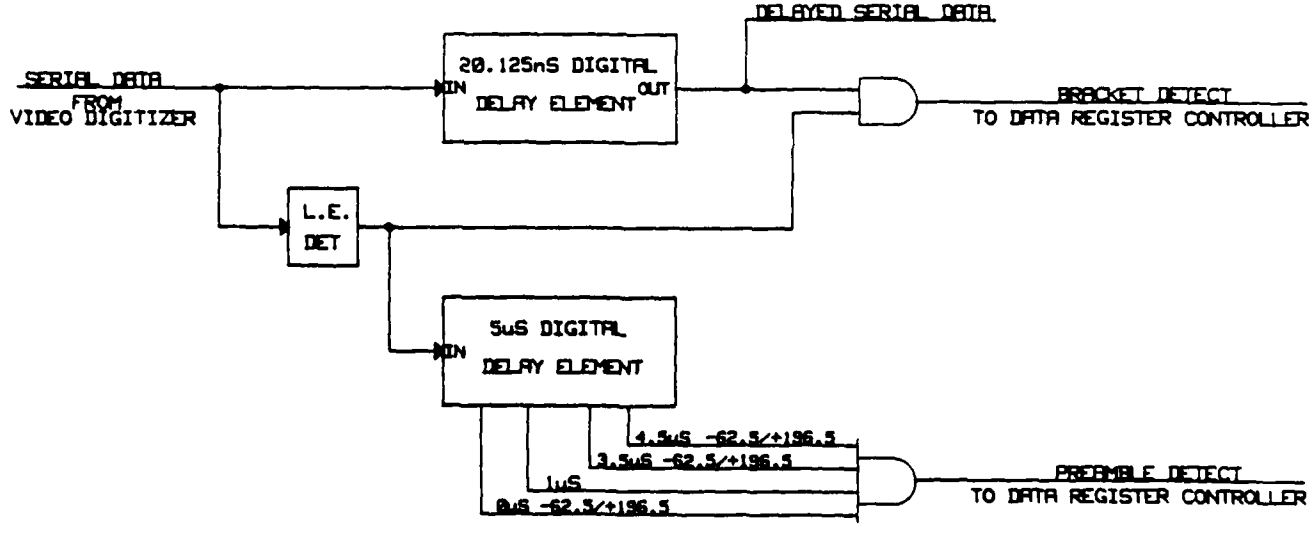


FIGURE 6.10.2.2.3-1. BRACKET AND PREAMBLE DETECTION LOGIC

6.10.2.2.4 Data Sampling Logic.

A simplified diagram of the Data Sampling Controller is presented in figure 6.10.2.2.4-1. It consists of an 11-bit address counter, a programmed ROM, an octal D-type register to eliminate spurious glitches, and a multiplexing circuit that selects different outputs from the ROM for sampling ATCRBS or Mode S (short and long) reply data.

The address counter is held at 0 until the Reply in Progress flip-flop (IC_1) is set. Once this flip-flop is set, the address counter starts cycling at the 16-MHz clock rate. As the address counter counts up, the sampling and load commands are generated from the ROM. Since the ROM provides both Mode S and ATCRBS commands, it is necessary to select the outputs. The sampling sequence is selected through multiplexer M_1. The multiplexer is controlled by the state of the Mode S/ATCRBS Detected flip-flop (IC_2). This flip-flop is set whenever the Reply in Progress flip-flop is set by the PREAMBLE DETECT pulse. When set, the Mode S data sampling sequence is selected. To determine if the Mode S reply is short or long, the first bit of the reply is sampled and the result is latched into the Long/Short flip-flop (IC_3). This flip-flop selects, via multiplexer M_3, the correct parity control signals from the ROM to control the Mode S reply parity decoder and the appropriate reply end command for Mode S replies.

When the reply end command is produced, all three flip-flops (IC_1, IC_2, and IC_3) are reset, terminating the data sampling sequence and forcing the address counter back to 0, ready for the next reply.

The data sampling logic controls the Mode S reply parity decoder, multiplexers, a 16-bit serial-to-parallel shift register, and a "load bit" counter used to keep track of the number of bits shifted into the register. This logic is presented in figure 6.10.2.2.4-2.

6.10.2.2.5 Reply Assembly.

The Reply Assembler State Controller formats the reply block and moves the data into the output buffer (FIFO). The State Controller accomplished this by monitoring four signals: (1) Reply Start (RPYSTRT), (2) A/DC Data Ready (A/D READY), (3) Reply Data Ready (DATARDY), and (4) Reply End (RPYEND).

For each reply received, the Downlink Receiver will transfer a block of 13 words to the STU computer. This block of data consists of reply time (range), antenna boresight azimuth, amplitude data (LOG SUM and LOG DELTA/LOG OMNI), phase bit (L/R), status word, and reply data bits. These data are held by different registers as shown in figure 6.10.2.2.5-1. Note that this logic consists of:

a. 20-bit Range Register
b. 14-bit Azimuth Register
c. 10-bit SUM Amplitude Sample plus Phase Bit Register
d. 10-bit DIFF/OMNI Amplitude Sample Register
e. 4-bit Reply Status Register
f. 16-bit Reply Data Register
g. 16-bit by 16-word Output Buffer (FIFO)
h. State Controller

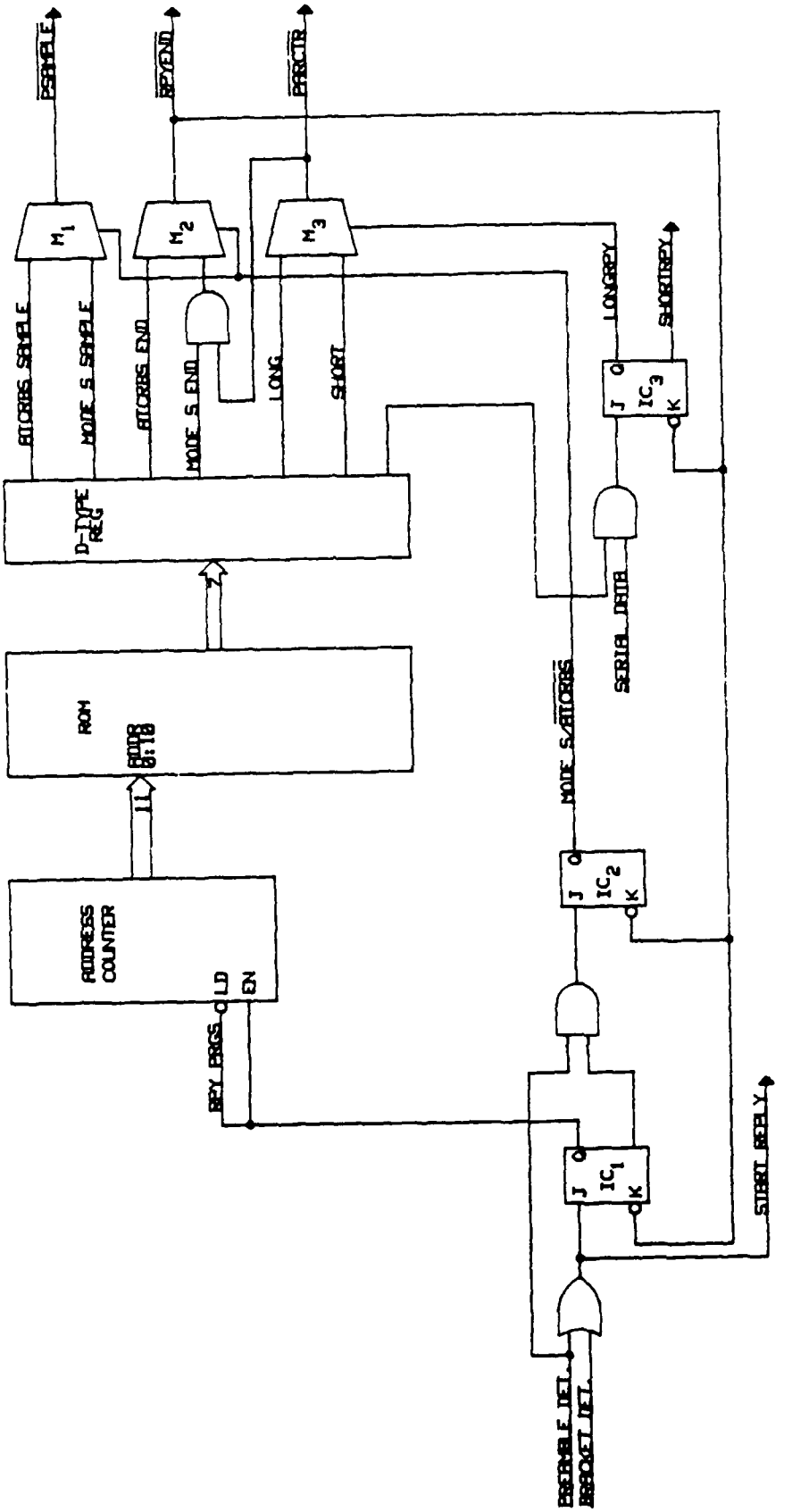


FIGURE 6.10.2.2.4-1. DATA SAMPLING CONTROLLER

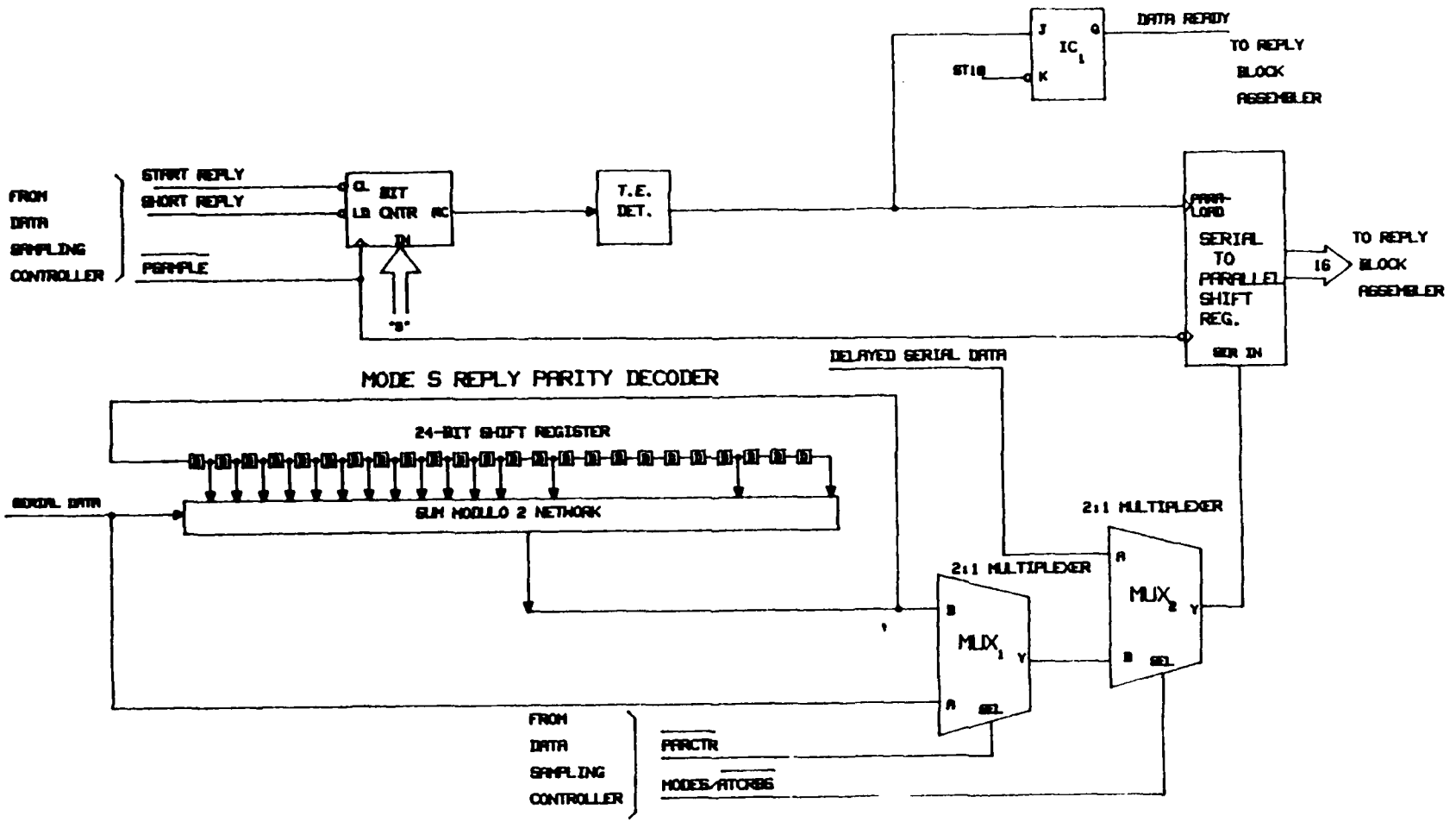


FIGURE 6.10.2.2.4-2. DATA SAMPLING LOGIC

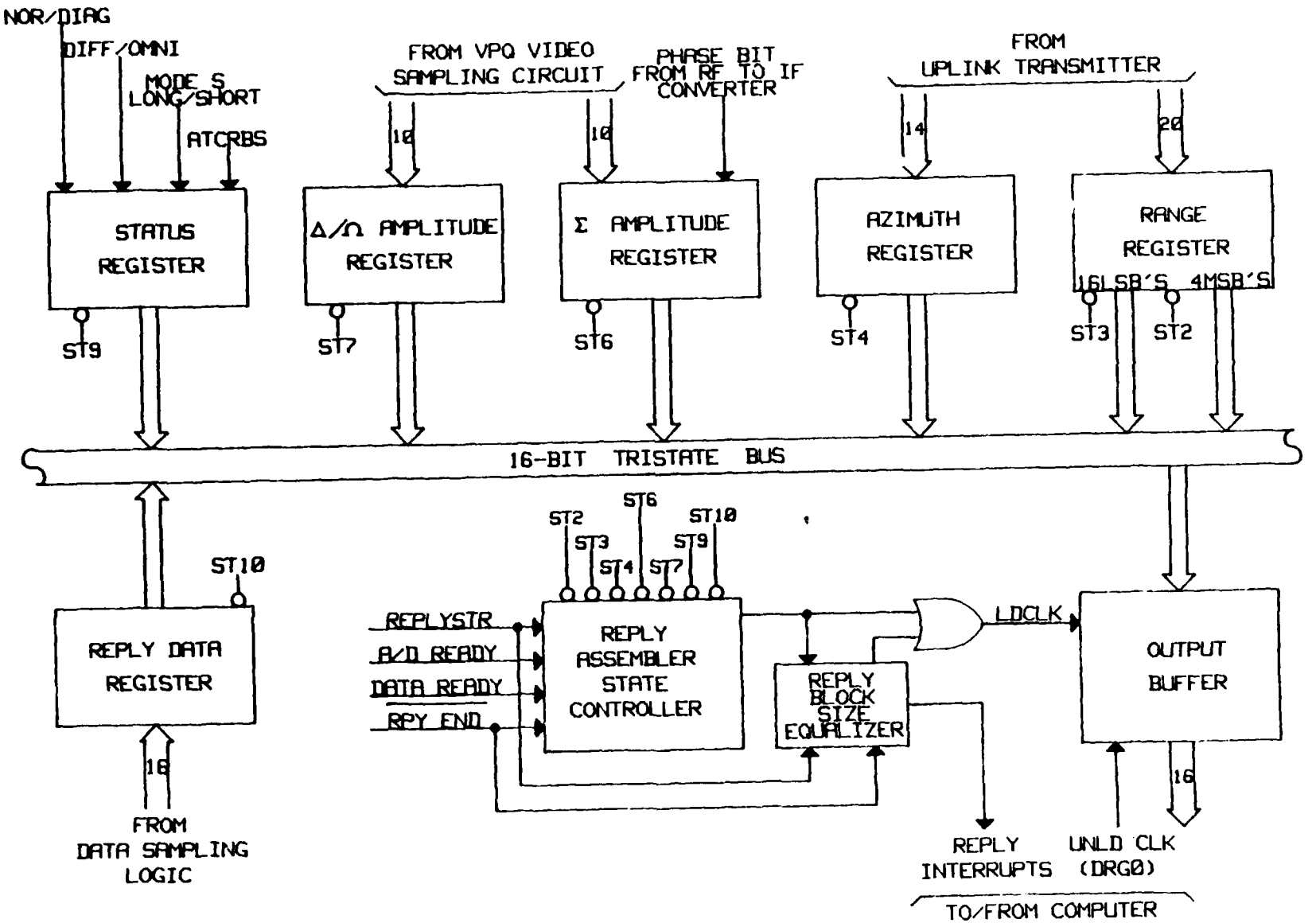


FIGURE 6.10.2.2.5-1. REPLY ASSEMBLY BLOCK DIAGRAM

The output of these registers are tied together forming a common tristate bus. Each register is then enabled by the State Controller, at the appropriate time, so that the data are transferred to the output buffer in the proper format. (Refer to the ARIES Hardware Maintenance Manual, Volume II, appendix D for Downlink Receiver reply block format.)

The Range Register receives its data from the Time-into-Transmission Counter located on the Uplink Transmitter board. Likewise, the Azimuth Register receives its data from the Azimuth Counter located on the same board. The data is latched into these registers at the time a reply bracket or preamble is detected.

The SUM Amplitude/Phase Detection Register and the DIFF/OMNI Amplitude Register receive their amplitude information from the VPQ board. The Phase Bit information is received directly from the RF to IF analog panel. The data is latched into these registers by the A/D READY signal received from the VPQ board.

The Status Register holds 4 bits of information defining the reply block. Two bits are used to specify the reply type: ATCBRS, Mode S Short, or Mode S Long. A third bit specifies whether DIFF or OMNI amplitude data is stored in the DIFF/OMNI Amplitude Register. Finally, the fourth bit specifies whether the reply was originally from the onboard simulation logic or from the ARIES reply generation circuitry. The status bits are latched into the Status Register at the same time the first reply word is loaded into the Reply Data Register.

The Reply Data Register receives its data from the 16-bit serial-to-parallel shift register. Data is loaded into this register after the Data Ready signal is raised by the Data Sampling Controller.

The operations of the State Controller may be understood by examining the state diagram of figure 5.10.2.2.5-2. State 0 is the initial state of the controller, entered whenever the Downlink Receiver is reset. It remains in this state until a reply bracket or preamble is detected causing the REPLYSTR signal to go high. This signal allows the controller to advance through states 2, 3, and 4, thereby transferring the range and azimuth data to the output buffer. When state 5 is reached, the controller waits for the end of conversion signal A/D READY to go high. This signal allows the controller to advance through states 6, 7, and 8, thereby transferring the LOG SUM amplitude and the LOG DIFF/LOG OMNI amplitude data to the output buffer. At state 8, the controller resets the Phase Bit and the A/D READY flip-flops.

The controller remains in state 8 until reply data becomes available (DATA READY goes high). Since it takes 16 μs to assemble 16 reply data bits, this wait state is needed. As soon as a 16-bit reply data word is ready, the controller advances through state 9, enabling the status word register to the output buffer, to state 10, enabling the data register to the output buffer. If the reply was ATCRBS, which contains only 16 data bits, reply processing is completed causing the RPYEND signal to go high. If the reply was Mode S, which contains 56 or 112 data bits, reply processing is not completed and the RPYEND signal remains low. This condition causes the controller to enter state 11 until additional reply data becomes available. States 10 and 11 are repeated until no further data is available causing the RPYEND signal to go high. This signal allows the controller to return to state 0 and the whole process is ready to repeat on the next reply.

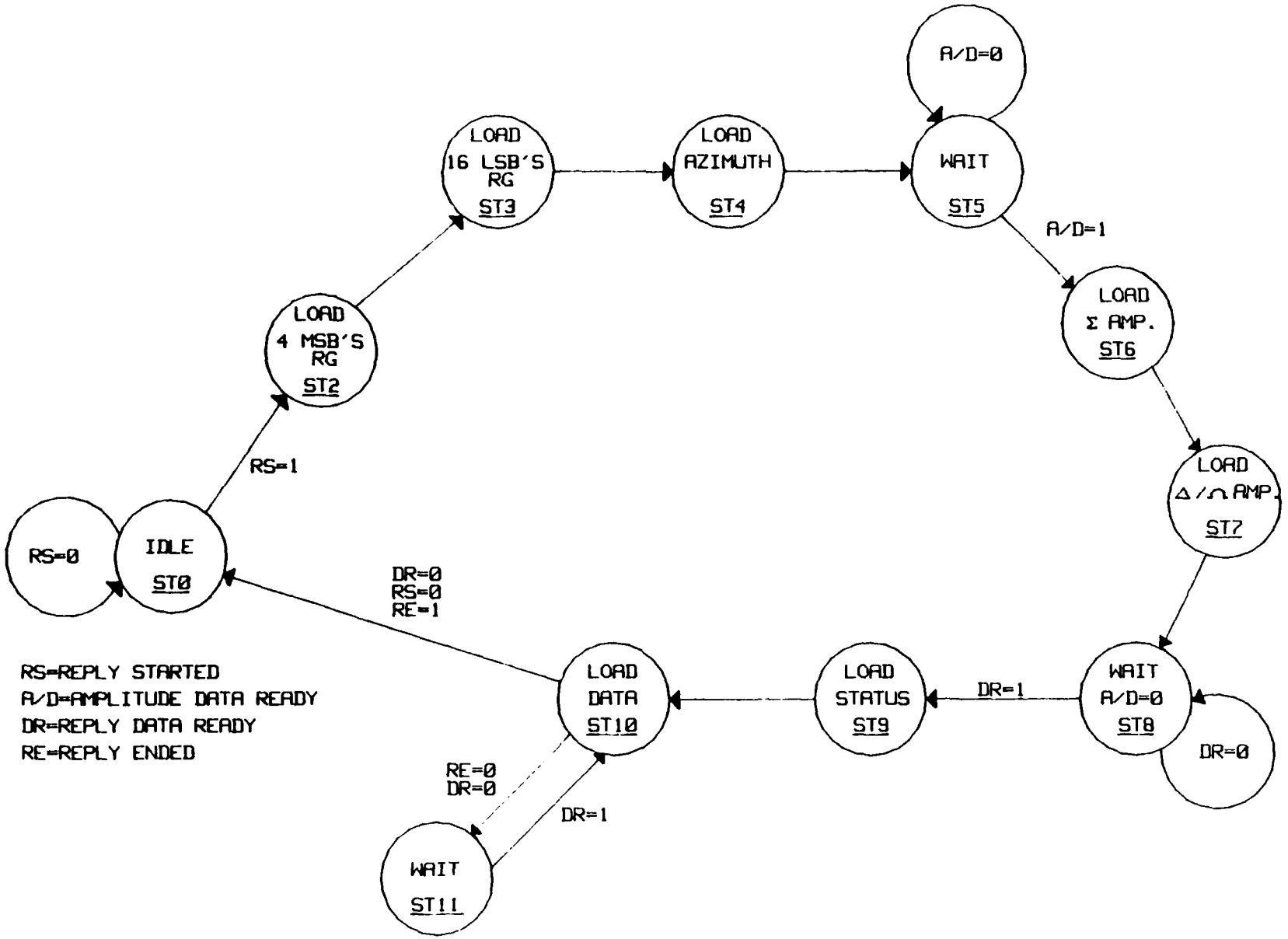


FIGURE 6.10.2.2.5-2. STATE DIAGRAM OF THE REPLY ASSEMBLE STATE CONTROLLER

To standardize the reply data block size to 13 words for any reply type (ATCRBS, Short Mode S, or Long Mode S), the logic shown in figure 6.10.2.2.5-3 was implemented. This logic loads the additional words after the State Controller completes its data transfers to the output buffer. The logic is initiated when a reply is detected (REPLYSTR = 1). This signal is leading edge detected, setting J/K flip-flops IC_1 and IC_2. At this point, the counter CNTR is preset to a value of 3 and activated. The counter is incremented, each time a word is loaded into the output buffer, to keep track of the number of words loaded. During this time, NAND gate A is inhibited by the /Q output of flip-flop IC_2.

Reply processing ends (RPYEND = 0) after all of the reply data is sampled and flip-flop IC_2 is reset enabling NAND gate A and AND gate B. With gate A enabled, the toggle flip-flop IC_3 is allowed to produce the remaining load clocks to complete the reply data block to its full 13-word size. Flip-flop IC_3 is prevented from toggling until the State Controller returns to state 0 (RPYEND = 1). If the reply assembled was a Long Mode S, the counter would hold a value of 15, at which time the ripple carry output would go high. However, if the reply assembled was an ATCRBS (Short Mode S), the counter would hold the value 9 (12). With less then 15 words loaded into the output buffer, flip-flop IC_3 begins to toggle producing additional load clocks transferring "don't care" words into the output buffer. After each load clock is produced, the counter is incremented until its maximum value is reached and the ripple carry goes high. Flip-flop IC_1 is reset deactivating the counter, along with gate A, preventing further generation of load clocks. This standardization is implemented to simplify the ISR used to fetch the reply blocks.

Also, J/K flip-flop IC_4 is set, which in turn enables gate B to pass a high signal to the Send Attention flip-flop IC_5. On the following clock, flip-flop IC_5 is set. With its /Q output fed back to the K-input of IC_4, gate B is disabled on the next clock cycle when IC_4 is reset. The output of flip-flop IC_5 then goes low one clock cycle later, producing the two-clock wide SATN1 pulse. This pulse initiates the interrupt request to the computer indicating that a reply block is available.

6.10.2.2.6 Reply Simulation Logic.

The reply simulation logic is shown in figure 6.10.2.2.6-1. This logic consists of a ROM, a ROM address counter, an octal D-type register, an 8:1 multiplexer, a 2:1 multiplexer, and some additional support logic.

The ROM is programmed with eight replies; four ATCRBS and four Mode S (two short and two long). These replies are listed in table 6.10.2.2.6-1. The output of the ROM is fed through the octal D-type register to filter out transient glitches generated each time the ROM address is updated. As the ROM address counter increments, the reply pulses are generated on all eight output lines of the ROM. Only one line is past through the 8:1 multiplexer controlled by a 4-bit binary counter referred to as the "reply selector."

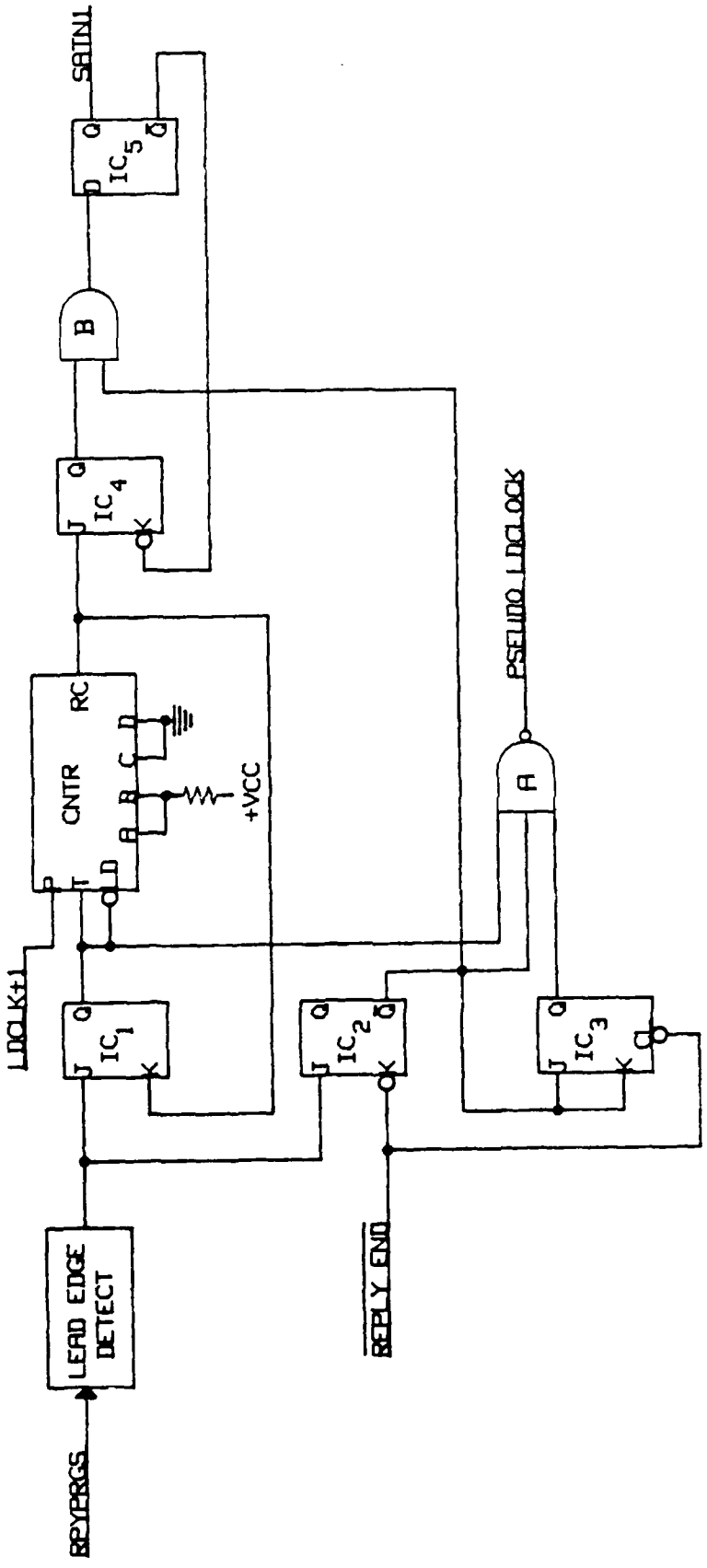


FIGURE 6.10.2.2.5-3. REPLY BLOCK SIZE EQUALIZER LOGIC

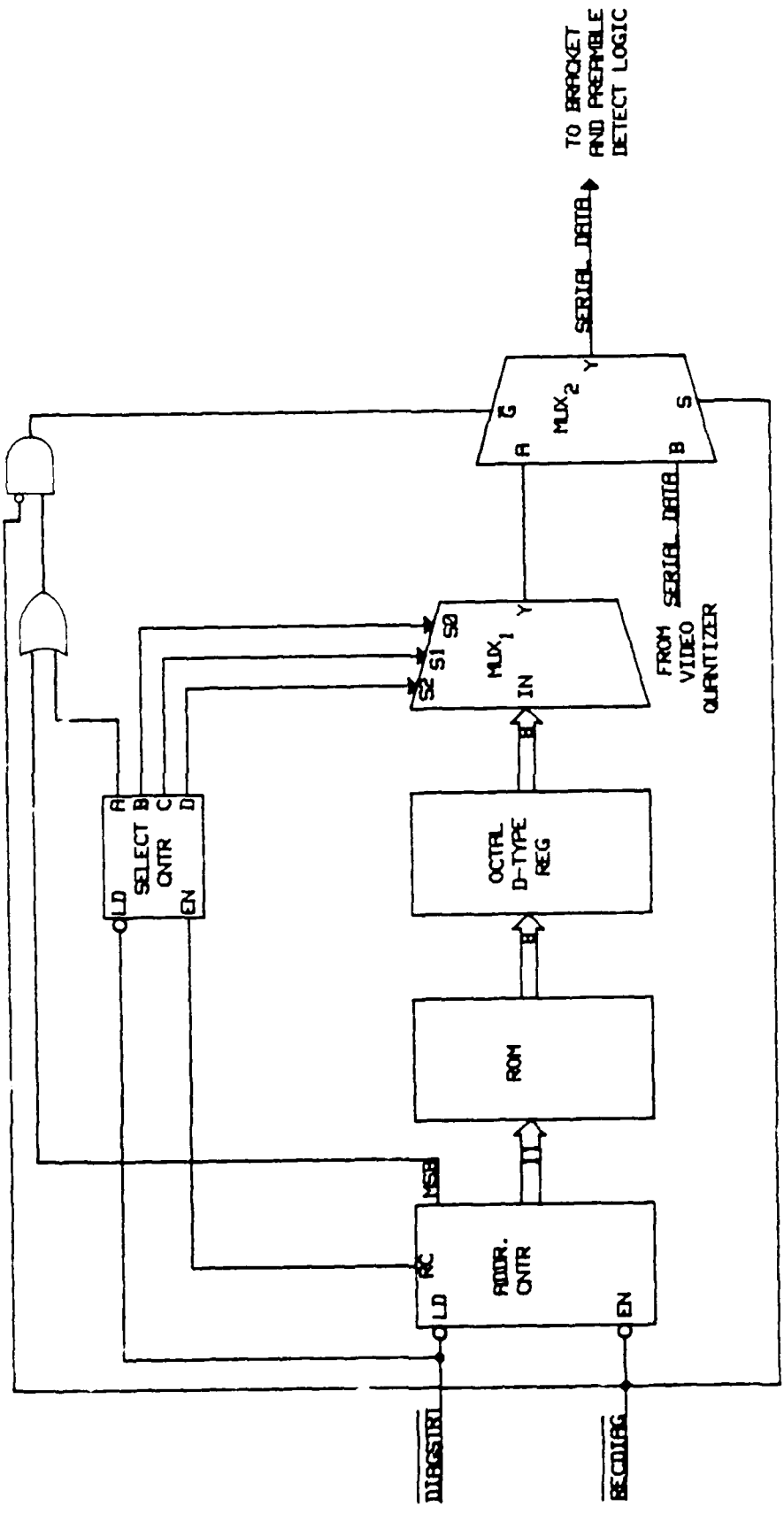


FIGURE 6.10.2.2.6-1. REPLY SIMULATION LOGIC

TABLE 6.10.2.2.6-1. ONBOARD SIMULATED REPLIES

REPLY TYPE	REPLY DATA
Mode S Short	27FF FFFF 0000 01
ATCRBS	FFFF
Mode S Short	2800 0000 FFFF F7
ATCRBS	8103
Mode S Long	A7FF FFFF FFFF FFFF FFFF FF00 0001
ATCRBS	AAAA
Mode S Long	A800 0000 0000 0000 0000 00FF FFF7
ATCRBS	D[illegible]5

When the Downlink Receiver board is placed in the diagnostic mode, control pulse $\overline{\text{DIAGSTRT}}$ is produced starting both the ROM address counter and the reply selecter at 0, and control signal RECON/$\overline{\text{REDIAG}}$ goes low enabling both the ROM address counter and the reply selector. Also, multiplexer MUX_2 selects the A-input as input to the receiver.

Note that this logic could produce a reply once every 128 μs. However, this reply rate could not be handled by the STU processor. Therefore, the reply generation rate was reduced to one reply every 512 μs. This was accomplished by taking the MSB of the ROM address counter and the LSB of the reply selector through a NAND function gate to produce an active low pulse through the first half of the ROM address loop every other address loop. This pulse is used to enable multiplexer MUX_2 once every four possible reply generations. To understand how this is possible, consider the following. The ROM address counter is a 12-bit counter but only 11 bits are needed to fully address the ROM. Therefore, the ROM is cycled twice for every complete address loop. The three MSBs of the reply selector are used to select one of eight possible replies through multiplexer MUX_1. This 3-bit field is updated once every two completed address loops allowing the LSB of the reply selector to toggle once every completed address loop.